

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

GABRIEL LUIS DE OLIVEIRA

**Desenvolvimento do sistema eletrônico de um  
veículo do tipo *Formula Student* elétrico  
baseado nas regras da Fórmula SAE**

Porto Alegre

2020

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

GABRIEL LUIS DE OLIVEIRA

**Desenvolvimento do sistema eletrônico de um veículo do  
tipo *Formula Student* elétrico baseado nas regras da  
Fórmula SAE**

Relatório Final do Projeto de Diplomação II  
apresentado ao Departamento de Engenharia  
Elétrica da Escola de Engenharia da Univer-  
sidade Federal do Rio Grande do Sul, como  
requisito parcial para Graduação em Enge-  
nharia Elétrica

Orientador: Prof. Dr. Tiago Roberto Balen

Porto Alegre

2020

GABRIEL LUIS DE OLIVEIRA

**Desenvolvimento do sistema eletrônico de um veículo do  
tipo *Formula Student* elétrico baseado nas regras da  
Fórmula SAE**

Relatório Final do  
Projeto de Diplomação II apresentado  
ao Departamento de Engenharia Elétrica  
da Escola de Engenharia da Universidade  
Federal do Rio Grande do Sul, como re-  
quisito parcial para Graduação em Enge-  
nharia Elétrica

---

**Prof. Dr. Tiago Roberto Balen**  
Orientador - UFRGS

---

**Prof. Dr. Luiz Fernando Ferreira**  
Chefe do Departamento de Engenharia  
Elétrica (DELET) - UFRGS

BANCA EXAMINADORA

---

**Prof. Dr. Tiago Roberto Balen**  
UFRGS

---

**Prof. Dr. Jeferson Vieira Flores**  
UFRGS

---

**Prof. Me. Igor Pasa**  
UFRGS

*À meus pais, Evandro e Marlete, e minha irmã Julia, por todo apoio durante minha  
jornada de vida e graduação.*

# Agradecimentos

Primeiramente, eu gostaria de agradecer à Universidade Federal do Rio Grande do Sul por me propiciar uma formação de excelência. Além de todos os professores e servidores da instituição, os quais foram vitais para o meu desenvolvimento ao longo do curso.

Agradeço à equipe e-Power UFRGS, que foi fundamental para a minha formação, proporcionando tanto experiências práticas com o desenvolvimento de projeto quanto a possibilidade de me fazer exercer um cargo de liderança. Vivência que me fez crescer como profissional.

Agradeço aos meus amigos e colegas, sempre dispostos a ajudar e me encorajar a seguir em frente no curso.

Por fim, agradeço também a Deus e minha família, que sempre acreditaram no meu potencial e me tornaram quem eu sou.

*O que sabemos é uma gota e o que ignoramos é um oceano.*

Isaac Newton

*Seu trabalho vai preencher uma grande parte da sua vida, e a única maneira de ficar realmente satisfeito é fazer o que você acredita ser um ótimo trabalho. E a única maneira de fazer um excelente trabalho é amar o que você faz.*

Steve Jobs

# Resumo

Além de estar associado à alta tecnologia pela indústria automotiva, o carro elétrico assumiu uma elevada posição na luta por um mundo ambientalmente sustentável. A previsão é que assuma de maneira significativa o mercado automotivo mundial, já que de acordo com o relatório da Bloomberg New Energy Finance (BNEF), os mesmos passem de 2 milhões para 56 milhões de unidades até 2040. Visando uma redução das emissões de dióxido de carbono e introdução de sistemas híbridos, a *SAE International - Society of Automotive Engineers* fundou em 2013 a categoria elétrica, conhecida por *Formula FSAE Electric*. No Brasil, a modalidade propiciou aos estudantes de engenharia a possibilidade de aplicar na prática os conhecimentos adquiridos, construindo um carro do estilo fórmula movido a eletricidade. Buscando desenvolver um sistema de baixo custo e que atenda aos requisitos impostos pela competição, o presente projeto visa projetar e testar a maior parte da eletrônica de baixa tensão do veículo, além de desenvolver o firmware da placa do diferencial. Passando a realizar, assim, um sistema composto por ECUs (*Engine Control Unit*), sensores, condicionadores de sinal e sistemas de segurança. Ao final, são apresentadas as dificuldades encontradas e, a partir delas, são sugeridos trabalhos futuros.

**Palavras-chave:** Eletrônica, *Engine Control Unit*, FSAE Brasil, veículos elétricos.

# Abstract

In addition to being associated with high technology by the automotive industry, the electric car has taken a high position in the fight for an environmentally sustainable world. They are expected to take a significant hold on the global automotive market, as according to the Bloomberg New Energy Finance (BNEF) report, they will rise from 2 million to 56 million units by 2040. Aiming at reducing carbon dioxide emissions and introducing hybrid systems, in 2013 SAE International - Society of Automotive Engineers founded the electric category, known as Formula FSAE Electric. In Brazil, the sport offered engineering students the possibility to apply their acquired knowledge in practice, building a formula-style car powered by electricity. Seeking to develop a low cost and energetically efficient system that meets the requirements imposed by the competition, the present project aims to design and test most low voltage electronics of the vehicle, in addition to developing the differential board firmware. Thus, a system composed of ECUs (Engine Control Unit), sensors, air conditioners signal and security systems. At the end, the difficulties encountered are presented and, based on them, future work is suggested.

**Keywords:** Electrical vehicles, Electronics, Engine Control Unit, FSAE Brazil.



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
<b>1.1</b>	<b>Propriedades gerais de veículos elétricos</b>	<b>10</b>
<b>1.2</b>	<b>A competição FSAE Brasil</b>	<b>12</b>
<b>1.3</b>	<b>A equipe e-Power UFRGS <i>Racing</i></b>	<b>14</b>
<b>1.4</b>	<b>Segurança em veículos elétricos</b>	<b>15</b>
<b>2</b>	<b>ESPECIFICAÇÕES TÉCNICAS</b>	<b>16</b>
<b>2.1</b>	<b>Veículos elétricos na fórmula SAE</b>	<b>16</b>
2.1.1	Baterias	16
2.1.2	Diferencial Eletrônico	18
<b>2.2</b>	<b>Sistemas eletrônicos</b>	<b>19</b>
2.2.1	Sensor da posição do pedal de aceleração (APPS - <i>Accelerator Pedal Position</i> )	19
2.2.2	Sensor de ângulo da direção (SWAS - <i>Steering Wheel Angle Position</i> )	20
2.2.3	Instrumentação da linha de freio	21
2.2.4	Interruptor de Inércia	23
2.2.5	Sistema de ultrapassagem do alcance do freio (BOTS - <i>Brake Over Travel System</i> )	23
2.2.6	Sistema de monitoramento do nível de isolamento (IMD - <i>Insulation Monitoring Device</i> )	25
2.2.7	Modo pronto para dirigir - <i>Ready to Drive Mode</i>	26
2.2.8	Sistema de checagem de plausibilidade do freio (BSPD - <i>Brake System Plausibility Device</i> )	26
2.2.9	Luz indicativa do sistema de tração (TSAL - <i>Tractive System Active Light</i> )	27
2.2.10	Circuito de desligamento - <i>Shutdown Circuit</i>	27
<b>2.3</b>	<b>Protocolo de comunicação CAN</b>	<b>29</b>
<b>3</b>	<b>DESENVOLVIMENTO DO HARDWARE</b>	<b>32</b>
<b>3.1</b>	<b>ECU - Unidade de Controle Eletrônico</b>	<b>33</b>
<b>3.2</b>	<b>Sensor do pedal de aceleração (APPS - <i>Accelerator Pedal Position</i>)</b>	<b>35</b>
<b>3.3</b>	<b>Sensor de ângulo da direção (SWAS - <i>Steering Wheel Angle Position</i>)</b>	<b>37</b>
<b>3.4</b>	<b>Modo pronto para dirigir - <i>Ready to Drive Mode</i></b>	<b>39</b>
<b>3.5</b>	<b>Instrumentação do pedal de freio</b>	<b>39</b>
<b>3.6</b>	<b>Sistema de Segurança</b>	<b>43</b>
3.6.1	Interruptor de Inércia	43
3.6.2	Sistema de checagem de plausibilidade do freio (BSPD)	45
3.6.3	Circuito de Set/Reset do sistema	48

3.6.4	Sistema de monitoramento do nível de isolamento . . . . .	49
3.6.5	Sistema de ultrapassagem do alcance do pedal de freio (BOTS) . . . . .	50
<b>3.7</b>	<b>Luz indicativa do sistema de tração (TSAL) . . . . .</b>	<b>51</b>
<b>4</b>	<b>DESENVOLVIMENTO DO FIRMWARE - DIFERENCIAL . . . . .</b>	<b>55</b>
4.0.1	Estado Inicial - <i>Init State</i> . . . . .	55
4.0.2	Estado de Comunicação - <i>Communication State</i> . . . . .	56
4.0.3	Estado de Erro - <i>Error State</i> . . . . .	56
<b>5</b>	<b>EXPERIMENTOS E RESULTADOS . . . . .</b>	<b>57</b>
<b>5.1</b>	<b>Experimentos com a Luz Indicativa do Sistema de Tração (TSAL) .</b>	<b>57</b>
5.1.1	Simulação . . . . .	57
5.1.2	Experimento em laboratório . . . . .	60
<b>5.2</b>	<b>Experimentos com o Sistema de Checagem de Plausibilidade do freio (BSPD) . . . . .</b>	<b>61</b>
5.2.1	Simulação . . . . .	61
<b>5.3</b>	<b>Experimentos com o firmware do diferencial . . . . .</b>	<b>63</b>
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>64</b>
<b>7</b>	<b>APÊNDICE A . . . . .</b>	<b>66</b>
<b>8</b>	<b>REFERÊNCIAS BIBLOGRÁFICAS . . . . .</b>	<b>84</b>

# 1 Introdução

## 1.1 Propriedades gerais de veículos elétricos

Em substituição aos veículos a combustão, a modalidade elétrica vem ganhando força ao longo dos anos e evidentemente é dotada de certa tecnologia, relacionada com a tração, que acaba recebendo maior enfoque. As principais características distintas desses carros, em relação ao modelo antigo, são: *drive-train*, *power-train* e forma de obtenção de energia.

- ***Drive-train*:** O *drive-train* de um veículo é o grupo de componentes responsáveis por transmitir a potência gerada no motor para as rodas. Na modalidade a combustão, com o uso de engrenagens, a chamada transmissão tem início no volante, passando pela embreagem, caixa de câmbio, eixo de transmissão e diferencial até chegar nas rodas. No caso dos veículos elétricos, a transmissão é simplificada pois eles adquirem o máximo torque do motor na rotação zero (ao contrário da combustão), o que leva a uma partida mais rápida e sem necessidade da troca de marchas e embreagem.
- ***Power-train*:** O *power-train* (figura 1), parte adicional somente em veículos elétricos, é constituído de inversores (responsáveis por fornecer a rotação adequada para os motores e recuperar a energia em caso de frenagem regenerativa), baterias e motores, os quais possuem um rendimento elevado se comparados com a modalidade antiga, chegando a cerca de 90% no caso elétrico e cerca de 30-40% no caso a combustão [26].

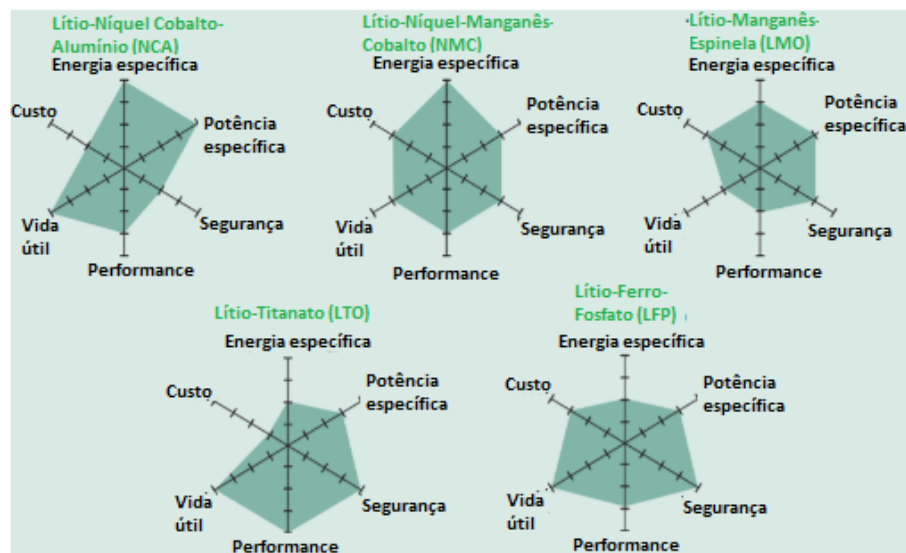
Figura 1 – Power-Train Nissan Leaf.



Fonte: Nissan.

- **Bateria:** Fonte de energia de todos os componentes do veículo elétrico. Apesar do seu surgimento girar em torno de 1800, antes da criação dos carros, passou a integrar os primeiros veículos em 1859. Sendo que as primeiras eram constituídas de chumbo. As principais encontradas no mercado atualmente, voltadas para a área automotiva, são constituídas de variações da de Íon-Lítio: NCA (Lítio-Níquel-Cobalto-Alumínio), NMC (Lítio-Níquel-Manganês-Cobalto), LMO (Lítio-Manganês-Espinela), LTO (Lítio-Titanato) e LFP (Lítio-Ferro-Fosfato). As mesmas possuem vantagens e desvantagens quando comparadas (como ilustrado na figura 2), ficando a cargo da montadora ou equipe de competição a escolha do tipo ideal.

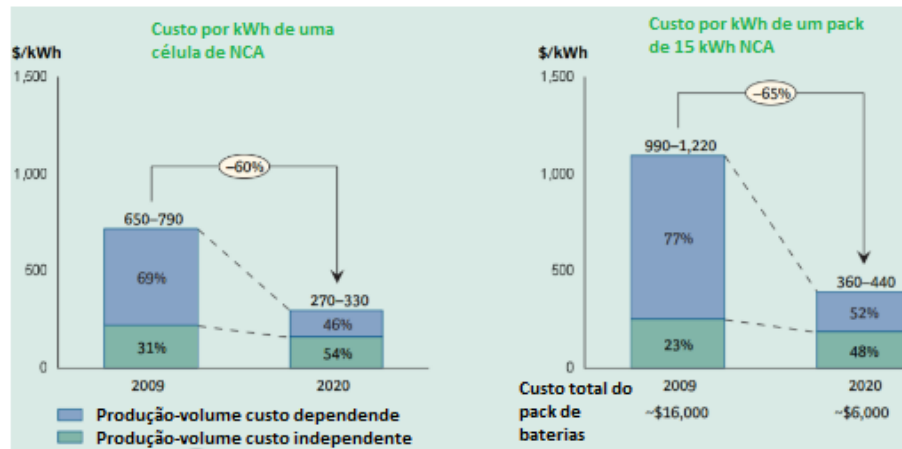
Figura 2 – Prós e contras das baterias derivadas do Lítio-Íon.



Fonte: Adaptado de BCG Research.

Uma das características mais impactantes da bateria nos carros elétricos atuais é a sua energia específica, que é definida pela razão entre a capacidade de armazenar energia e seu peso. Em comparação com a gasolina utilizada nos modelos a combustão, a energia específica das baterias é apenas 1% da energia dos combustíveis, o que ocasiona um limite nas distâncias percorridas pelos veículos devido ao peso adicional. Segundo a BCG, as baterias chegam a ocupar cerca de 20 a 25% do peso total dos carros elétricos de porte pequeno atuais [14]. Um dos fatos que também impacta no projeto dos carros atuais e na competição da FSAE é o custo das baterias. Atualmente é a parte mais cara do veículo devido a quantidade de células usadas, porém, ao longo dos anos esse mesmo preço tende a sofrer uma queda devido a tendência de maior produção e tecnologia envolvida.

Figura 3 – Preço das baterias de NCA de 2009 a 2020.



Fonte: Adaptado de BCG Research.

A figura 3 apresenta uma estimativa comparando os anos de 2009 e 2020, a qual indicou uma redução no preço de cerca de 60% para uma bateria do tipo NCA [14].

## 1.2 A competição FSAE Brasil

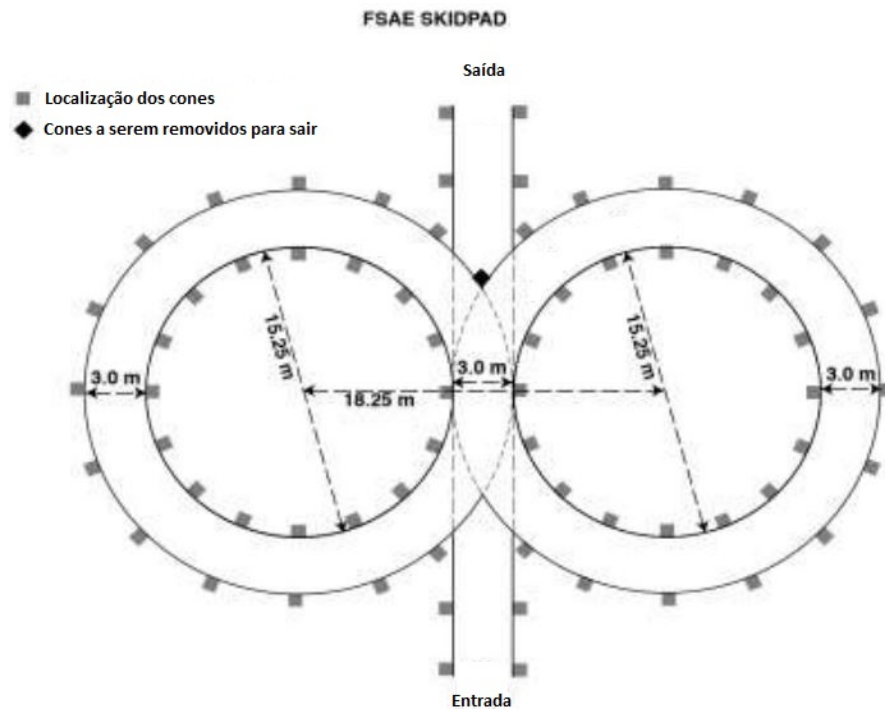
A fórmula SAE BRASIL é uma competição nacional, criada em 2004, que tem por objetivo desafiar os estudantes a construir um carro de corrida do tipo fórmula para competir em uma série de provas, tanto estáticas quanto dinâmicas. Inicialmente com a modalidade a combustão, a FSAE inclui hoje a modalidade de carros elétricos.

A competição consiste em cinco eventos dinâmicos e três estáticos, cada um com a sua própria pontuação. As equipes são julgadas baseadas em seu desempenho em relação às outras equipes. O vencedor de cada evento recebe o máximo de pontos atribuídos para a mesma e as equipes com pior desempenho recebem uma porcentagem dos pontos restantes. Os eventos dinâmicos visam testar o desempenho dos carros de corrida, com foco na aceleração, frenagem e manuseio, já os estáticos avaliam as decisões da equipe com relação à engenharia do carro. As provas podem ser divididas em:

**Aceleração:** Testa a aceleração do veículo em linha reta percorrendo uma pista de 75m. A equipe que concluir em menor tempo recebe a pontuação máxima.

**Skidpad:** Os condutores completam duas voltas em uma pista que consiste em dois círculos formando uma figura de oito (figura 4), sendo que somente a segunda volta é cronometrada. Tem como objetivo testar a capacidade de realizar curvas do veículo.

**Autocross:** Testa a capacidade de realizar manobras e o desempenho do manuseio do carro em um percurso que possui inúmeras curvas fechadas e obstáculos. Em teoria,

Figura 4 – Evento de *SkidPad*.

Fonte: Adaptado de FSAE 2020 *Rules*.

o carro com a maior combinação de aceleração, frenagem e capacidade de realização de curvas recebe a maior pontuação.

**Enduro:** O evento de resistência, também conhecido por enduro, tem como objetivo testar a durabilidade e a confiabilidade geral de um carro em um longo percurso. A eficiência energética do mesmo é medida ao longo da pista utilizando um medidor dado pela competição, podendo estimar assim, o grau de eficiência das baterias utilizadas e se o veículo não ultrapassa a potência e velocidade máxima permitida, segundo a fórmula SAE.

**Apresentação de negócios:** O evento reforça a ideia de que a equipe está desenvolvendo um protótipo de carro destinado ao mercado. As equipes precisam apresentar seu conceito na frente de um painel de potenciais investidores, com foco na identificação do mercado, previsão de vendas e rentabilidade.

**Design:** Consiste em uma série de discussões em pequenos grupos em que os membros da equipe devem apresentar a engenharia por trás de seu carro a um grupo de especialistas do setor. As categorias julgadas variam entre as competições, mas geralmente podem ser divididas em subequipes: aerodinâmica, chassi, eletrônica, ergonomia, gerenciamento da equipe, powertrain e suspensão.

**Análise de custo:** Antes da competição, cada equipe envia um documento contendo o custo para cada componente em seu carro. A decomposição de custos inclui

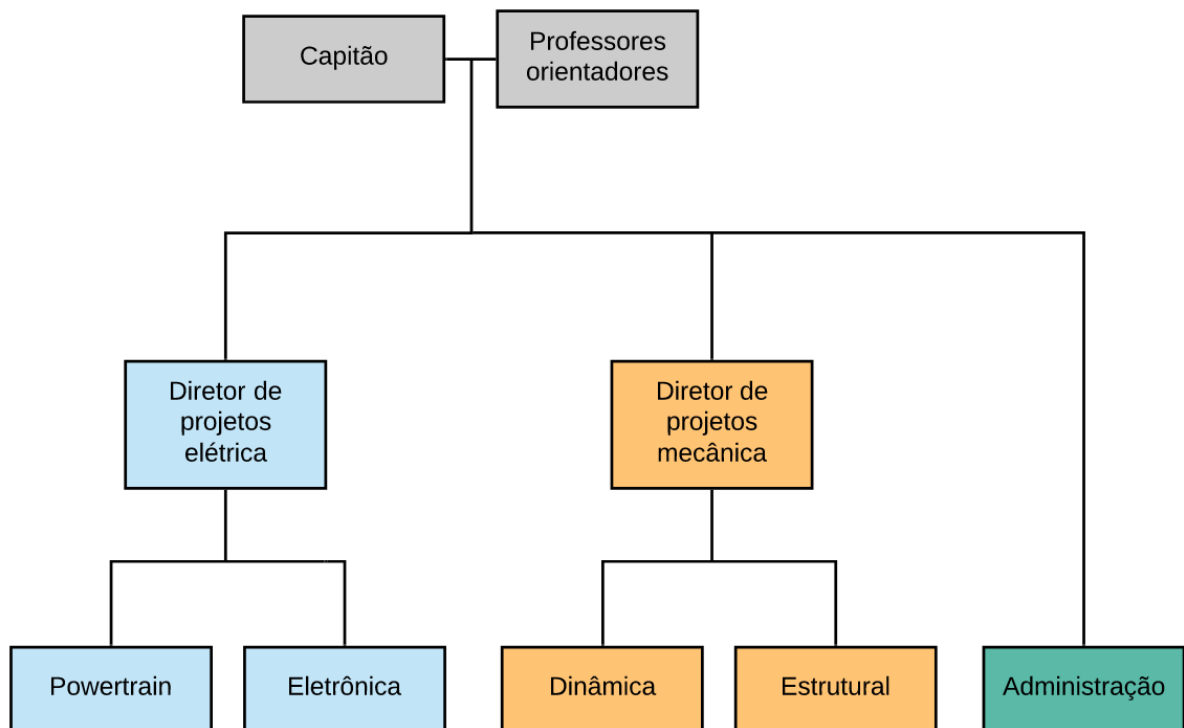
também os custos associados à sua fabricação e montagem. Na competição, cada equipe justifica suas decisões de engenharia do ponto de vista financeiro.

### 1.3 A equipe e-Power UFRGS *Racing*

Em 2018, os alunos da Universidade Federal do Rio Grande do Sul fundaram a primeira equipe da modalidade de carros elétricos da instituição, a e-Power UFRGS *Racing*, tendo como característica relevante o projeto do próprio motor elétrico. A equipe participou da competição em 2018 e 2019, porém sem um projeto totalmente concluído. O mesmo será desenvolvido no presente ano.

A equipe é organizada em áreas, duas pertencentes a elétrica (eletrônica e powertrain), duas a mecânica (dinâmica e estrutural) e uma administrativa. Em cada área pode-se ter também sub áreas. Cada área técnica possui um diretor de projetos e todos os membros são aconselhados pelos professores orientadores. Essas características são apresentadas no organograma da figura 5.

Figura 5 – Organograma da equipe.



Fonte: Elaborado pelo autor.

A área que suportou o desenvolvimento do presente trabalho foi a da eletrônica.

## 1.4 Segurança em veículos elétricos

O desenvolvimento de um carro do estilo fórmula elétrico tem diversos desafios associados a ele, um dos mais importantes são as medidas de segurança adequadas, de forma a garantir o bem estar do motorista. Um veículo desse tipo deve ter um bom desempenho para atingir altas velocidades em curtos períodos de tempo, mantendo a integridade de todos os componentes.

Como os carros da fórmula SAE são projetados por estudantes de engenharia e não profissionais do ramo automobilístico, diversos sistemas de segurança tanto mecânicos quanto elétricos devem estar presentes. Entre eles, pode-se destacar: Controle de nível de isolamento entre o barramento de alta tensão, baixa tensão e chassi, desligamento dos circuitos diante de acelerações bruscas, identificação de falhas devido a sensores assumindo valores inadequados, redundância de componentes, chaves gerais de acionamento e botões de emergência. Além dessas estratégias, uma Análise de Modos de Falhas e Efeitos (FMEA) deve ser entregue junto com a documentação do veículo para a competição. A mesma permite uma análise de todas as possíveis falhas no sistema elétrico e como a equipe trata cada uma delas, de forma a diminuir significativamente as probabilidades de danos ao piloto e veículo.

O objetivo deste trabalho é desenvolver, portanto, alguns módulos do sistema de segurança do veículo:

- Sistema de checagem de plausibilidade do freio;
- Circuito de Set/Reset do sistema;
- Sistema de monitoramento do nível de isolamento do sistema de tração;
- Sistema de ultrapassagem do alcance do pedal de freio;
- Luz indicativa do sistema de tração;
- Interruptor de inércia.

Além disso, foi feito o desenvolvimento do hardware e firmware de uma das ECU's do carro, a ECU do diferencial.



## 2 Especificações técnicas

A seguir, será feita uma breve revisão sobre os veículos elétricos e seus principais componentes, conceitos básicos dos sensores que serão utilizados na competição e do protocolo de comunicação escolhido. Além disso, será descrito brevemente sobre essas tecnologias nos carros atuais e como outras equipes da fórmula SAE abordam esses conceitos.

### 2.1 Veículos elétricos na fórmula SAE

No caso da presente competição, utilizando como base as normas da competição, o time e-Power UFRGS se propôs a construir um veículo do tipo fórmula. De acordo com o descrito nos capítulos anteriores, pode-se listar certos componentes do carro, os quais são relevantes para o desenvolvimento da parte de baixa tensão do presente trabalho. Nesse caso, será abordado sobre o armazenamento de energia e o diferencial eletrônico.

#### 2.1.1 Baterias

O fornecimento de energia dos veículos da competição será dado através de 2 bancos de baterias, uma para o sistema de baixa tensão e outra para o de alta. A de alta tensão é constituída por células de  $\text{LiNiCoMnO}_2$  com tensão nominal de 3,7 V, 31 Ah e capacidade de descarga de 1C, isto é, podem descarregar com uma corrente de 31A instantâneo. A de baixa é constituída pela mesma liga citada anteriormente, porém com uma capacidade de fornecimento de 2,5 Ah.

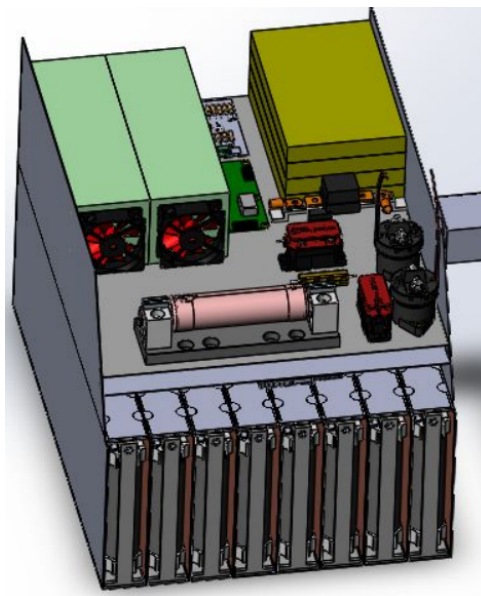
No caso do sistema de alta tensão, existe um conjunto de 8 segmentos em série, utilizando a configuração 12s1p, isto é, 12 células em série e 1 em paralelo, formando um segmento. Assim, a associação gera uma tensão nominal de 355,2 V ( $8 \times 12 \times 3.7\text{V}$ ) e uma capacidade de fornecimento de 31 Ah ( $1 \times 31\text{Ah}$ ).

O sistema de baixa tensão vai operar com 4 células em série e uma em paralelo, obtendo uma tensão equivalente de 14,8 V e capacidade de fornecimento de 2,5 mAh.

Na figura 6 é ilustrado como será feita a construção da caixa do acumulador com os 5 segmentos da bateria no andar inferior e a maioria dos componentes de potência no andar superior.

De acordo com as regras da competição o acumulador das baterias deverá ser protegido por pelo menos dois relés, fazendo com que o sistema de alta tensão abra em

Figura 6 – Projeto do acumulador.



Fonte: Equipe e-Power UFRGS

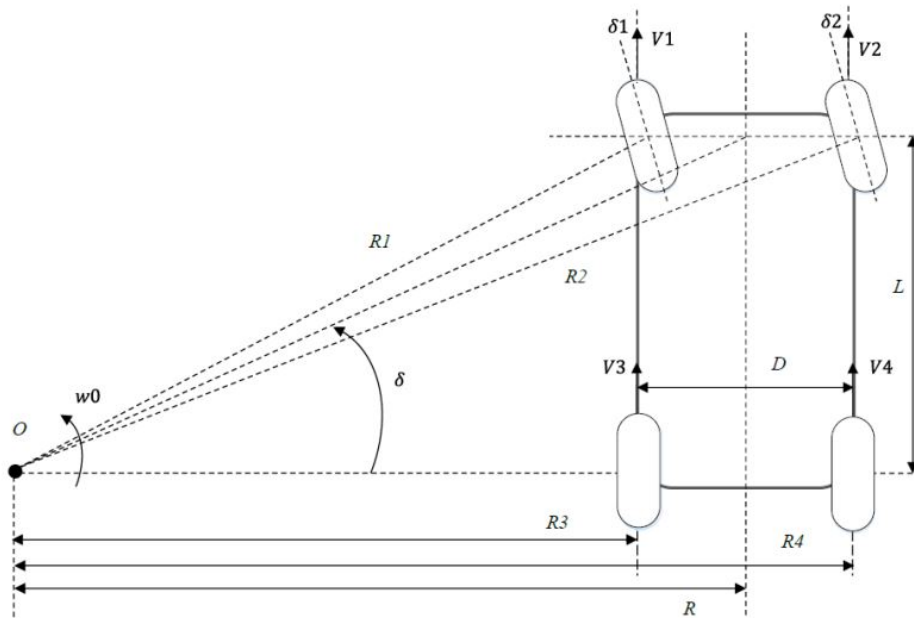
caso uma de falha detectada por algum sensor. Ademais, um fusível para proteção do sistema contra sobre correntes deverá ser instalado nesse mesmo circuito.

### 2.1.2 Diferencial Eletrônico

O diferencial, item presente na maioria dos veículos, sucintamente pode ser descrito como um dispositivo responsável por controlar a rotação de cada roda, fazendo com que diante de uma curva a roda mais exterior a curva adquira uma velocidade de rotação maior se comparada com a interior. Um aumento da velocidade de uma, ocasiona em uma redução da oposta. Por exemplo, quando um veículo realiza uma curva à esquerda, a roda deste lado possuem uma menor velocidade linear (percorrem uma circunferência menor) do que a do lado direito (figura 7).

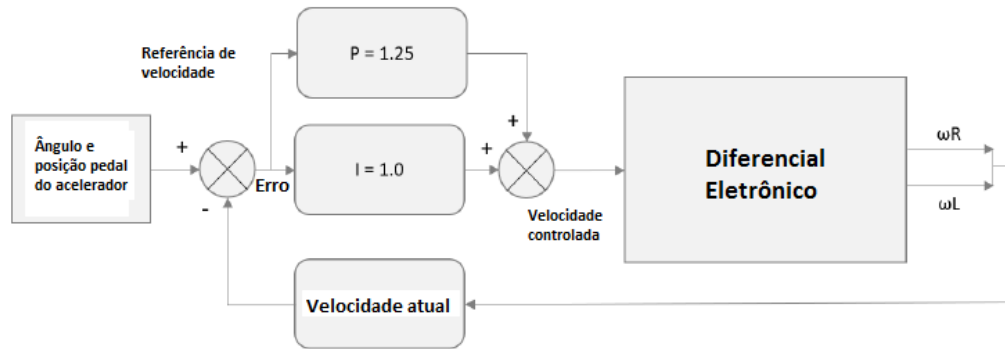
Tal dispositivo, quando empregado em veículos competitivos, pode ser implementado de forma elétrica ou mecânica. No caso da presente equipe, um protótipo de um diferencial já fora realizado [11], o qual a partir da aquisição do sinal do ângulo da direção, posição do pedal do acelerador e rotação atual dos motores, usa um laço de controle com um controlador PI para obter qual deverá ser a rotação de cada um dos dois motores traseiros do carro, para qualquer situação de operação. O diagrama de blocos desse sistema é ilustrado na figura 8. Esse dispositivo acabou servindo de base para o desenvolvimento do diferencial eletrônico atual.

Figura 7 – Veículo generalista realizando curva à esquerda.



Fonte: Adaptado de *A Study on the Control Performance of Electronic Differential System for Four-Wheel Drive Electric Vehicles*.

Figura 8 – Diagrama de blocos do diferencial eletrônico.



Fonte: Adaptado de *Electronic Differential for Electric Vehicles*[11].

A equação utilizada para modelar o diferencial é a de Ackermann-Jeantand, apresentada na equação 2.1.

$$\omega = \omega_{atual} \left[ \frac{L \pm (D/2) \cdot \tan(\theta)}{L} \right] \quad (2.1)$$

Onde  $L$  é a distância entre as rodas frontal e traseira,  $D$  a distância entre eixos,  $\omega_{atual}$  a velocidade desejada para o veículo e  $\theta$  o ângulo da direção. O modelo aplicado à roda esquerda é apresentado na equação 2.2 e aplicado à roda direita na equação 2.3.

$$\omega_{left} = \omega_{atual} \left[ \frac{L + (D/2) \cdot \tan(\theta)}{L} \right] \quad (2.2)$$

$$\omega_{right} = \omega_{atual} \left[ \frac{L - (D/2) \cdot \tan(\theta)}{L} \right] \quad (2.3)$$

## 2.2 Sistemas eletrônicos

O presente capítulo visa fundamentar a respeito dos sistemas eletrônicos que serão desenvolvidos no presente trabalho, apresentando características gerais dos mesmos, requisitos impostos pela FSAE e exemplos de aplicações no setor competitivo estudantil.

### 2.2.1 Sensor da posição do pedal de aceleração (APPS - *Accelerator Pedal Position*)

O pedal do acelerador passou por diversas mudanças ao longo do tempo, a versão mecânica que era acionada a partir de um eixo foi substituída pela versão eletrônica, sendo que o processamento desse sinal é realizado pela central eletrônica do veículo. Essa

moderna forma de aquisição de sinais foi utilizada primeiramente na fórmula 1 pela equipe McLaren, o que garantiu um melhor controle e instrumentação se comparado com a versão clássica.

O sensor em si tem por objetivo a geração de um sinal elétrico proporcional a pressão exercida pelo motorista sobre o pedal. O dispositivo geralmente encontrado nos veículos consiste de dois potenciômetros (garantindo uma redundância por motivos de segurança) atuando como divisores resistivos. Nesse caso, uma variação de posição do pedal do acelerador acarreta em uma variação de resistência do potenciômetro, e consequentemente uma variação de tensão na saída.

Outro tipo de sensor empregado pelas equipes é o sensor de posição do corpo da borboleta (TPS - figura 9). O mesmo era utilizado em motores a combustão para medir o ângulo de abertura da borboleta, variável utilizada pela ECU para o cálculo da quantidade de combustível a ser injetada, levando em conta a quantidade de ar admitido. No caso dos veículos elétricos da competição, esse sensor é mecanicamente fixado no pedal do acelerador, de forma a mensurar a pressão exercida pelo piloto sob o mesmo.

Figura 9 – Sensor TPS.



Fonte: Adaptado de DS.

A FSAE impõe o uso de 2 sensores com funções de transferência distintas. A intenção é que um possível curto-circuito que ocorra no circuito faça com que as saídas dos sensores sejam iguais somente quando o pedal não for pressionado. Além disso, o regulamento impõe que a integridade desse sinal deve ser garantida em todos os estágios de processamento do sinal até chegar nos inversores.

### 2.2.2 Sensor de ângulo da direção (SWAS - *Steering Wheel Angle Position*)

O sensor de ângulo da direção é responsável por fornecer a posição atual da direção de acordo com a requisição do piloto. Uma das opções de dispositivo frequentemente empregada é o tipo digital de sensor de ângulo, também chamado de *contact-less sensor*, o qual a partir da emissão de uma luz de *LED* e medição da taxa de interrupção da

mesma com um dispositivo óptico, gera uma forma de onda quadrada cuja frequência é proporcional a velocidade de giro da direção.

Um tipo de sensor analógico também utilizado frequentemente funciona de acordo com o princípio indutivo, sendo instalado na válvula de controle da direção eletro-hidráulica do veículo, fazendo a aquisição da velocidade de giro do volante e enviando esse sinal para a ECU. Por redundância, mais de um sensor se faz necessário na maioria dos casos.

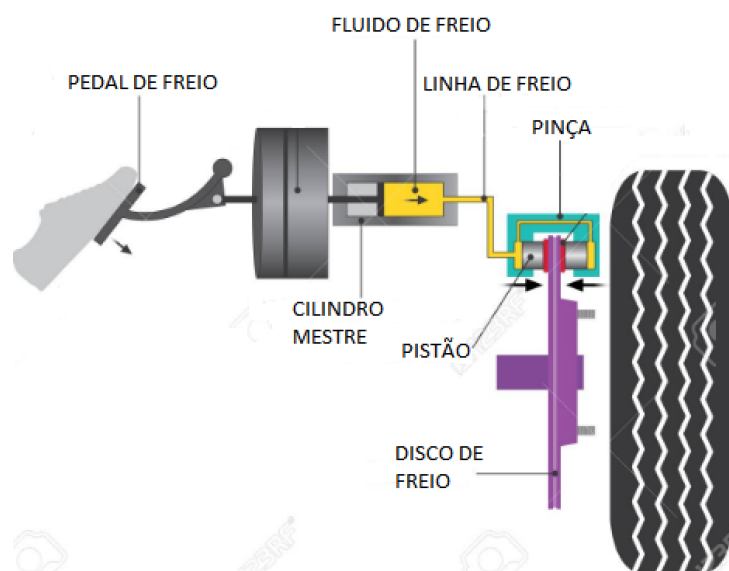
Geralmente no caso competição da fórmula SAE, o mesmo é montado em um local adequado na coluna de direção do veículo, fazendo com o sensor rotacione de acordo com o giro da direção, fato que acaba gerando uma tensão contínua a ser adquirida.

Não somente para atuar em conjunto com o pedal do acelerador para estimar a velocidade de rotação de cada roda, o ângulo de direção e sua taxa de incremento é vital, por exemplo, para garantir um controle de tração adequado, estratégica vital nas competições automotivas modernas.

### 2.2.3 Instrumentação da linha de freio

O sistema de freios automotivo é um dos itens que mais afetam a segurança na hora de conduzir o veículo, junto com a suspensão e a direção. No caso da presente equipe, a mesma faz uso de um sistema de freios hidráulico, ilustrado na figura 11. Primeiramente a pressão exercida pelo piloto sob o pedal é responsável pela movimentação do cilindro mestre e consequentemente, o fluido de freio. Esse fluido percorre a linha de freio até chegar no pistão. O mesmo é responsável pelo contato da pinça com o disco de freio, e assim frear a roda. Visando uma segurança adicional, o regulamento impõe o uso de dois cilindros mestres, um para cada conjunto de rodas. Desta forma, tem-se duas linhas de freio.

Figura 10 – Diagrama sistema de freio hidráulico.



Fonte: O autor.

Em linhas de freio hidráulico geralmente se faz uso de sensores de pressão, de forma a medir e posteriormente validar o sistema. Geralmente são empregados sensores que retornam uma tensão proporcional a pressão da linha (figura 11).

Figura 11 – Sensor de pressão da linha de freio.



Fonte: Adaptado de Fueltech.

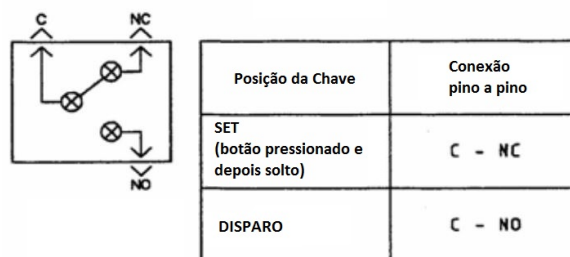
No caso da formula SAE, muitas equipes utilizam esse sensor para obter a pressão exercida pelo piloto sob o pedal de freio, processando e utilizando tal sinal por diversos fins, até mesmo como acionamento da luz de freio.

### 2.2.4 Interruptor de Inércia

Fundamental para o sistema de segurança dos veículos e obrigatório pelo regulamento da FSAE, o interruptor inercial é responsável por desativar o sistema de tração de veículos elétricos ou bloquear o funcionamento da bomba elétrica de combustível (cortando a sua alimentação) em veículos a combustão. Ou seja, objetiva proteger o veículo e seus usuários das consequências de uma frenagem brusca ou possível batida. É localizado geralmente nas portas, sobre o banco do motorista ou ao lado do motor.

Muitos sensores desse gênero funcionam a partir de um princípio mecânico-analógico, onde uma massa de prova fica confinada em uma gaiola com molas. Um choque em qualquer um dos 3 eixos do sistema de coordenadas acarreta no movimento da massa em relação a gaiola. Dependendo da intensidade do impacto a posição de uma chave é alterada de NF (normalmente fechada) para NA (normalmente aberta) (figura 12), possibilitando que o sistema de segurança do carro atue.

Figura 12 – Diagrama sensor inercial.



Fonte: Adaptado de First Technology.

Sensores comerciais possibilitam uma calibração para atuarem com acelerações de 8 a 30 vezes a aceleração da gravidade.

### 2.2.5 Sistema de ultrapassagem do alcance do freio (BOTS - *Brake Over Travel System*)

O BOTS é um sistema geralmente presente em competições automotivas com objetivo de desligar o sistema de tração do veículo quando o pedal de freio ultrapassa a faixa nominal de atuação.



Dependendo do tipo de vazamento ou entrada de ar no sistema de frenagem dos veículos de corrida, os mesmos podem ocasionar na falha desse sistema, o que ocasiona um risco para o condutor e veículo. Dessa forma, um sensor de pressão do tipo *push-pull* pólo único (por exemplo o da figura 13) é adicionado atrás do pedal de freio do carro.

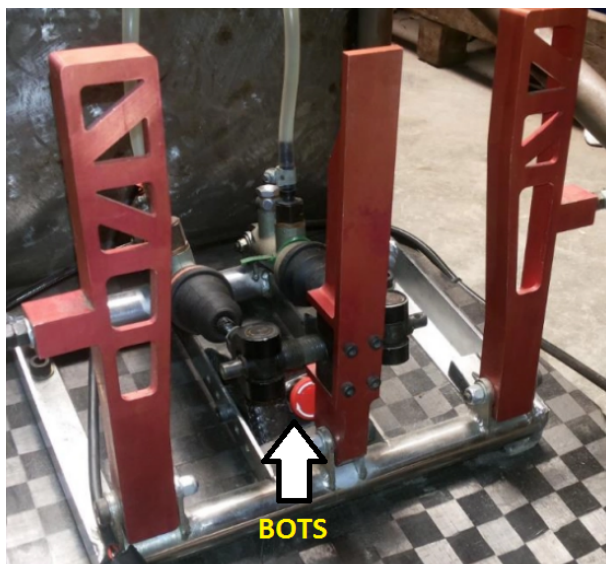
Se o sensor atuar significa que o sistema de freios está comprometido, portanto, o sistema de tração do veículo deverá ser desligado.

Figura 13 – Botão de pressão - BOTS.



Fonte: Adaptado de India Mark.

Figura 14 – *Pedal box* com o sensor BOTS localizado atrás do pedal central de freio.



Fonte: Adaptado de *Orion Racing India*.

O *pedal box* da equipe de formula SAE Orion Racing India é ilustrado na figura 14. Pode-se observar o sensor de pressão BOTS localizado atrás do pedal central de freio.

### 2.2.6 Sistema de monitoramento do nível de isolamento (*IMD - Insulation Monitoring Device*)

Visando uma proteção contra curto-circuitos, sistemas de monitoramento de isolamento são utilizados para a medição de diferença de impedância entre dois pontos, como entre o condutor de fase e a terra. Sensores de isolamento são frequentemente empregados na indústria, área médica, geração de energia e em sistemas automotivos.

Na área de veículos elétricos competitivos, seu uso se faz presente para evitar possíveis curto-circuitos entre o sistema de alta (tensões superiores a 25V AC ou 60V DC) e baixa tensão (chassi do veículo). O princípio de funcionamento do mesmo consiste em fazer uma aquisição da tensão entre dois pontos a cada intervalo de tempo, se a resistência entre os mesmos for menor do que a mínima especificada pela competição o sistema de tração do veículo deverá ser desacionado.

No caso específico da fórmula SAE, o sensor a ser utilizado é o Bender A-ISOMETER iso-F1 IR155-3203, -3204 ou equivalente IMD aprovado para o uso automotivo (figura 15). Quando uma falha de isolamento oriunda de uma fuga de corrente ocorrer, um led indicativo vermelho deverá ser acionado no cockpit, de forma a informar ao piloto a origem da desativação do sistema de tração do veículo. O respectivo sensor a ser utilizado possui duas saídas informativas. A primeira fica em nível lógico um nas condições normais de operação e assume o nível lógico zero quando um curto-circuito ou mal funcionamento do sensor for detectado. A segunda é uma saída *PWM* que tem por objetivo informar a resistência atual entre os sistemas, condições de sobtensão e falhas no próprio sensor.

Figura 15 – Bender A-ISOMETER iso-F1 IR155.



Fonte: Adaptado de Bender.

Por questões de segurança, o reinício de operação do carro, pós desativação do sistema, somente ocorrerá a partir do pressionamento de um botão de reset localizado na parte traseira, longe do motorista.

### 2.2.7 Modo pronto para dirigir - *Ready to Drive Mode*

Visando uma proteção adicional ao veículo, a FSAE inclui no regulamento que o carro somente deve responder ao pedal de aceleração quando o condutor estiver preparado, o que acaba evitando possíveis respostas do carro em situações como manutenção ou quando o piloto se encontra do lado de fora do cockpit. Dessa forma, o veículo se encontra no *Ready to Drive Mode*, ou modo pronto para dirigir, quando o sistema de tração estiver ativo, o pedal de freio estiver pressionado e um botão de pressão (indicado como *ready to drive no cockpit*) for pressionado. Após isso, um som de no mínimo 80 dBA deve ser reproduzido, entre 1 a 3 segundos de duração. A intensidade sonora é requerida em dBA para se certificar que o som a ser reproduzido se encontrará dentro da banda audível.

A maioria das equipes utiliza, portanto, um *buzzer*, o qual funciona a partir do princípio piezoelétrico, ou seja, aproveita a capacidade de certos cristais gerarem diferenças de pressão mecânica em resposta a uma diferença de potencial aplicada em seus terminais.

### 2.2.8 Sistema de checagem de plausibilidade do freio (BSPD - *Brake System Plausibility Device*)

O BSPD, ou sistema de checagem de plausibilidade do freio, é imposto pela organização da fórmula SAE para assegurar que o piloto não pressione o pedal de freio de forma brusca e uma corrente de amplitude elevada esteja sendo fornecida da bateria aos motores. Se o mesmo atuar, os relés de proteção da bateria devem abrir.

Na presente competição, o circuito responsável por esse sistema deverá ser não programável e *standalone*, isto é, operar de forma independente de outro hardware ou software. Geralmente o mesmo é implementado por um circuito comparador com histerese analógico, o qual gera uma saída de tensão positiva se o sinal a ser comparado ultrapassa certo limiar.

Por questões de segurança, o reinício de operação do carro, pós desativação do sistema, somente ocorrerá a partir do pressionamento de um botão de reset localizado na parte traseira do carro, longe do motorista.

### 2.2.9 Luz indicativa do sistema de tração (TSAL - *Tractive System Active Light*)

Durante a competição, tanto durante as provas quanto no carregamento da bateria é importante ter uma noção visível do estado do sistema de tração, principalmente por questões de segurança. Ou seja, saber se o sistema está ligado ou quando está operando com uma alta tensão.

Visando tal fato a fórmula SAE impõe que os carros possuam uma luz indicativa, a qual deverá ser vermelha piscante sempre que o sistema de tração esteja com uma alta tensão e verde caso contrário. Um exemplo dessa iluminação, empregado pela equipe de *Formula Student* de Zürich, é ilustrada no topo do carro na figura 16.

Figura 16 – Exemplo de TSAL equipe AMZ.



Fonte: Adaptado de *AMZ Racing*.

Um impasse de tal regra da competição é de que o sistema não pode conter um controle por software, ou seja, a tensão do sistema de tração deverá ser medida diretamente. Diante disso muitas equipes utilizam conversores DC-DC do tipo *buck*, de forma a reduzir as altas tensões para níveis passíveis de uso.

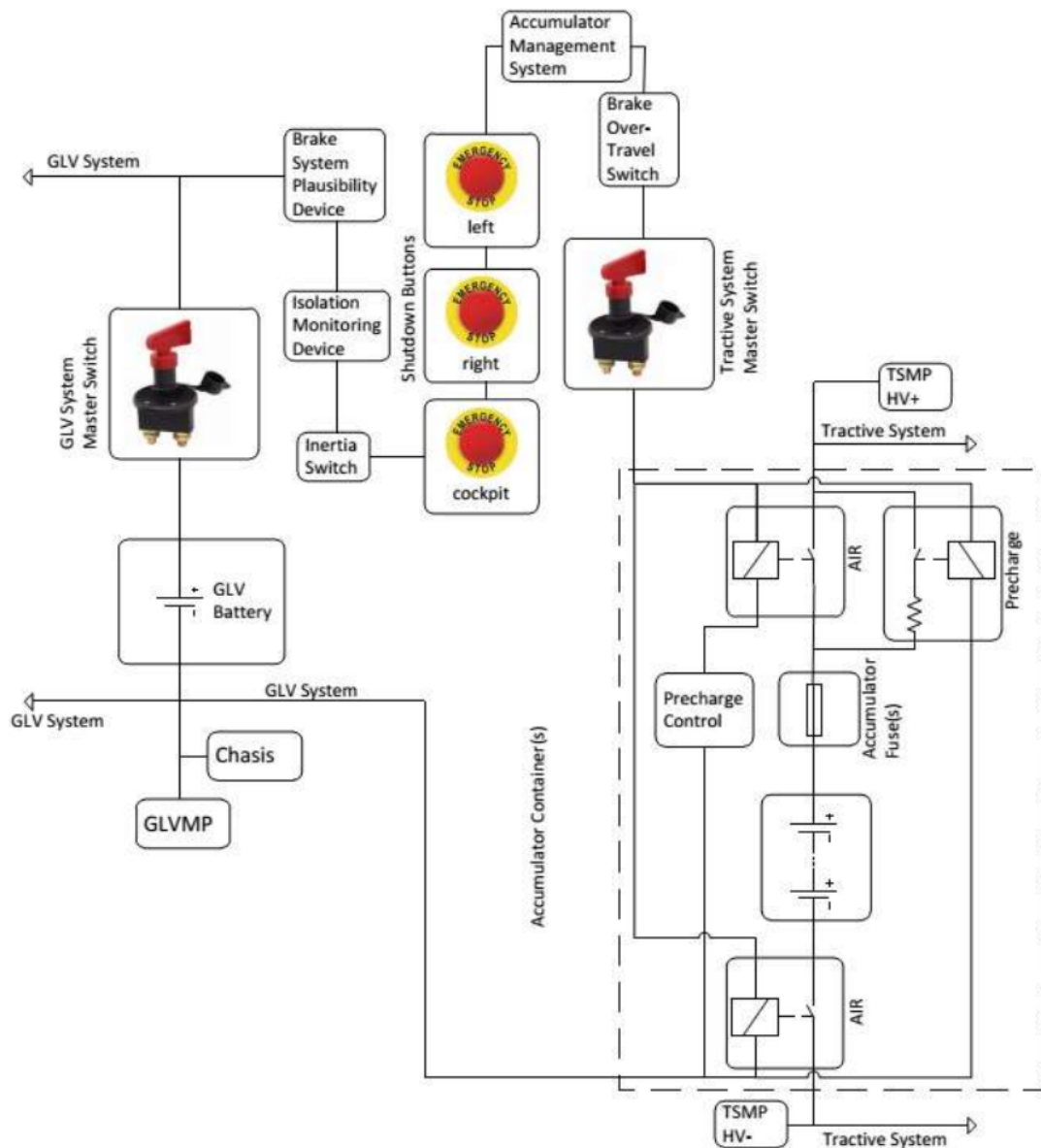
### 2.2.10 Circuito de desligamento - *Shutdown Circuit*

O circuito de desligamento do veículo é o responsável pela atuação dos relés de potência da bateria (AIRs - *Accumulator Isolation Relays*) em caso de falha de algum sistema elétrico ou quando algum membro da equipe julgar necessário. Quando o mesmo atuar, o barramento de alta tensão do veículo deverá ser desenergizado em menos de 5

segundos, de forma a garantir que o time possa realizar eventuais manutenções no carro de forma segura.

O circuito de desligamento é ilustrado na figura 17. Nele é possível observar que a alimentação de todo o circuito provém da bateria de baixa tensão, a qual possui terminal negativo aterrado no chassi do veículo. Esse circuito de proteção está em série com os relés da bateria de alta tensão, sendo acionado pelo circuito de controle de pré-carga (*Precharge Control*) e desacionado pelo circuito de descarga (*Discharge circuit*). Ambos circuitos são utilizados para evitar picos de corrente devido a transições bruscas de tensão no acionamento dos motores. Os mesmos foram implementados pela equipe. O presente trabalho possui foco nos demais circuitos e formas de atuação do piloto no sistema de segurança do carro.

Figura 17 – Circuito de desligamento do veículo.



Fonte: Adaptado de *FSAE Rules 2020*.

Na figura pode-se destacar também as duas chaves *master switch* (responsáveis pelo acionamento e desacionamento do circuito de desligamento) e os botões de *shutdown*, formas tanto do piloto quanto da equipe atuar no circuito de segurança.

## 2.3 Protocolo de comunicação CAN

Controller Area Network (CAN) é um protocolo de comunicação desenvolvido pela empresa alemã BOSCH em 1986, com foco na indústria automotiva. Tipicamente interliga

uma rede de módulos ou nós usando dois condutores (geralmente par trançado) [1]. Possui características relevantes, como maior confiabilidade de transmissão das informações e alta imunidade a interferências eletromagnéticas, quando comparado à protocolos padrões como o UART, SPI ou I2C.

O mesmo faz uso do conceito multi-master, onde todos os módulos acoplados ao barramento podem se tornar mestres em certo momento e escravos no outro. Todos esses dados são transmitidos pela comunicação conhecida por *broadcast*, onde apenas um remetente envia grandes blocos de dados para todos os outros, no caso do CAN com uma taxa máxima de 1 megabit por segundo (Mbit/s). As mensagens que são transmitidas podem ser agrupadas em 4 tipos. *Data Frame*, *Remote Frame*, *Error Frame* e *Overload Frame*. O *Data Frame* carrega dados do transmissor para o receptor. O *Remote Frame* é transmitido pelo barramento quando é necessária a transmissão de um *Data Frame* com o mesmo identificador. O *Error Frame* é transmitido por uma unidade quando algum tipo de erro no barramento é identificado. O *Overload Frame* é utilizado para adicionar um tempo extra entre a transmissão de um *Data Frame* ou *Remote Frame* [3].

O padrão de bits de uma mensagem padrão CAN (*Standard CAN*) é ilustrado na figura 18 e descrito posteriormente:

Figura 18 – Mensagem padrão CAN.



Fonte: *Introduction to the Controller Area Network (CAN)* [1].

**SOF - Single dominant start of frame:** Bit indicador de início de mensagem e de sincronismo das mesmas.

**Bit Identifier:** Conjunto de bits que estabelecem a prioridade das mensagens, quando menor o valor maior a prioridade.

**RTR - Remote Transmission Request:** Bit indicador de quando determinada informação é requisitada por determinado nó.

**IDE - Identifier Extension:** Bit indicador que a mensagem transmitida não possui extensão adicional.

**r0 - Bit reservado.**

**DLC - Data Length Code:** Determina o tamanho da mensagem a ser transmitida (em bytes).

**0...8 Bytes Data:** Mensagem a ser transmitida.

CRC - *Cyclic Redundancy Check*: Contém a soma dos bits que serão transmitidos, usado para detecção de possíveis erros.

ACK - *Acknowledges*: Bit indicador que a mensagem recebida anteriormente estava corrompida e a nova será a correta.

EOF - *End of Frame*: Bit indicador do final da mensagem.

IFS - *Interframe Space*: Contém o tempo necessário para que o controlador mova corretamente uma mensagem recebida para a sua posição apropriada.

As redes que fazem uso desse protocolo operam com dois condutores, os quais possuem os sinais de dados denominados *CAN\_HI* e *CAN\_LO*. Para definir o nível lógico de cada bit é avaliada a diferença de potencial entre *CAN\_HI* e *CAN\_LO*. Ao contrário do que ocorre com a notação comum, na rede CAN o bit recessivo é identificado como “1” (nível lógico alto) e o dominante como “0” (nível lógico baixo). Assim, só obtém-se o bit "1" quando existe uma diferença de potencial entre *CAN\_HI* e *CAN\_LO*.

Devido as vantagens mencionadas, o protocolo de comunicação CAN foi o escolhido para ser aplicado no veículo da equipe.

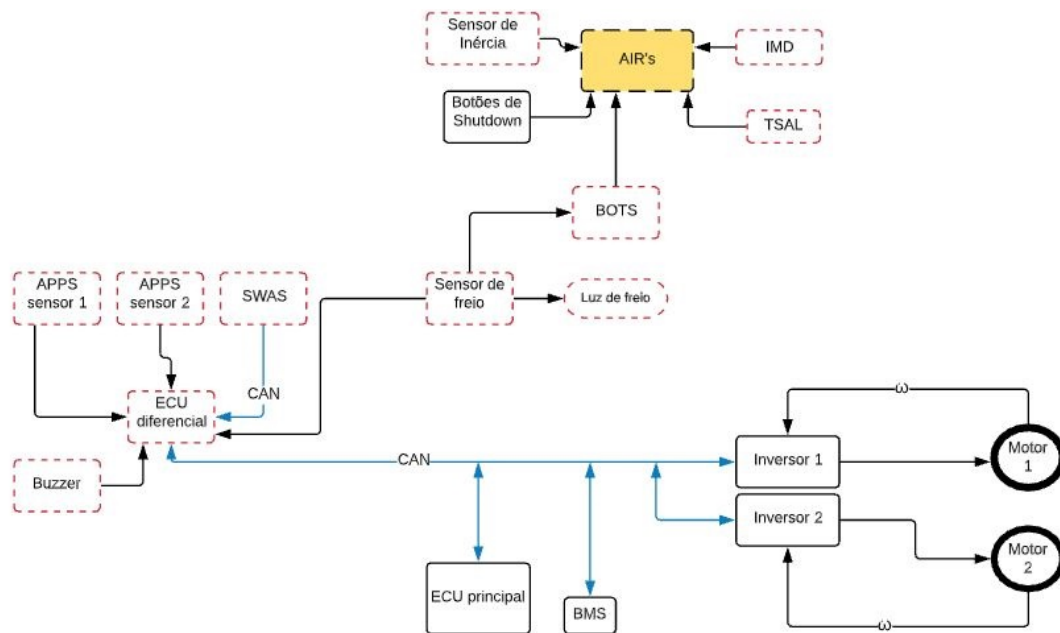


### 3 Desenvolvimento do Hardware

Com base na fundamentação apresentada anteriormente, será descrito a seguir como será implementado cada sistema introduzido nesses capítulos. Utilizando como base principal as normas da competição.

Um diagrama funcional de todo o sistema elétrico do carro, mostrando as interfaces relevantes, é ilustrado na figura 19.

Figura 19 – Diagrama elétrico do veículo.



Fonte: Elaborado pelo autor.

Nesse diagrama percebe-se o fluxo de dados entre os sistemas do veículo relevantes para este trabalho, principalmente os pontos onde é utilizada a comunicação CAN (linhas em azul). As demais conexões serão analógicas. Vale ressaltar que os módulos com linhas pontilhadas são os que serão abordados no presente trabalho.

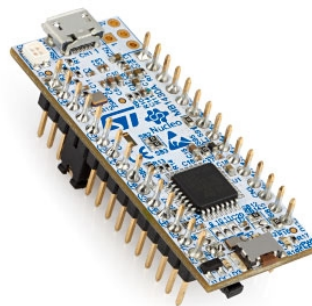
O sistema como um todo foi desenvolvido modularmente, usando assim conectores de alimentação e sinal entre as placas de circuito impresso. Os circuitos foram confeccionados em placas de fenolite universais, de forma a testar o funcionamento de todos os circuitos no carro antes da produção definitiva. Os módulos que serão abordados no presente trabalho serão descritos nos próximos sub-capítulos, sendo os demais desenvolvidos pela equipe.

### 3.1 ECU - Unidade de Controle Eletrônico

A ECU (*Electronic Control Unit*) ou Unidade de Controle Eletrônico possui a função de lidar tanto com a aquisição de sinais quanto como o processamento e atuação sobre o veículo. Nesse presente caso, um microcontrolador será encarregado de tais funções.

O dispositivo selecionado foi a placa de desenvolvimento ARM STM32 Nucleo-32, devido a capacidade de processamento da mesma, número de entradas analógicas (visando futuras telemetrias), protocolo CAN integrado e o fato da equipe receber elas como parte de um patrocínio. O processador é um Arm Cortex M4, possuindo 128 Kb de memória flash e 32 Kb de SRAM. Possui dois conversores ADCs de 12 bits (com frequência de amostragem configurada para 1,41 MHz), 4 comparadores, 3 amplificadores operacionais, 4 conversores Digital/Analógico, referência de tensão interna e clocks em tempo real. O kit de desenvolvimento será utilizada para fins de teste, posteriormente a equipe utilizará somente os microprocessadores, confeccionando as próprias placas de circuito impresso. A placa do kit de desenvolvimento é ilustrada na figura 20 e uma renderização da placa que será confeccionada pela equipe na figura 21.

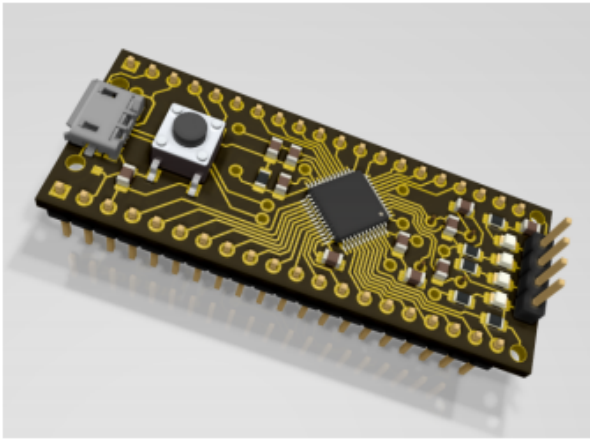
Figura 20 – Placa de desenvolvimento com STM32G431.



Fonte: *ST Microelectronics*.

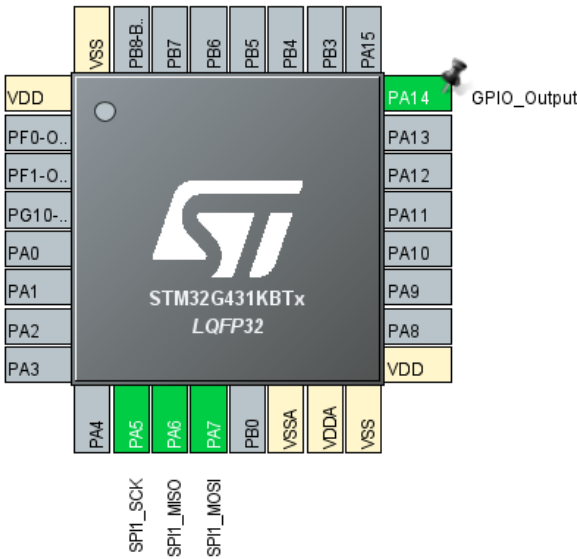
A implementação do firmware de cada ECU do veículo será feita utilizando a IDE (Ambiente de Desenvolvimento Integrado) STM32Cube, uma ferramenta gráfica que possui uma intuitiva interface para a configuração do microcontrolador via linguagem de programação C. Um exemplo de atribuição do pino PA14 como GPIO (General Purpose Input/Output) é demonstrado na figura 22.

Figura 21 – Placa de desenvolvimento confeccionada pela equipe.



Fonte: Elaborada pelo autor.

Figura 22 – STM32Cube visualização de pinagem.



Fonte: Elaborada pelo autor.

### 3.2 Sensor do pedal de aceleração (APPS - *Accelerator Pedal Position*)

O sensor que foi utilizado para instrumentar o pedal do acelerador é o de posição TPS (*Throttle position sensor*) ilustrado na figura 23. O mesmo será mecanicamente encaixado no pedal, de forma a variar a tensão de saída proporcionalmente a pressão exercida pelo piloto sob o pedal. Ele possui uma tensão de alimentação de 5V, com uma variação de 0,6V a 4,6V (aproximadamente).

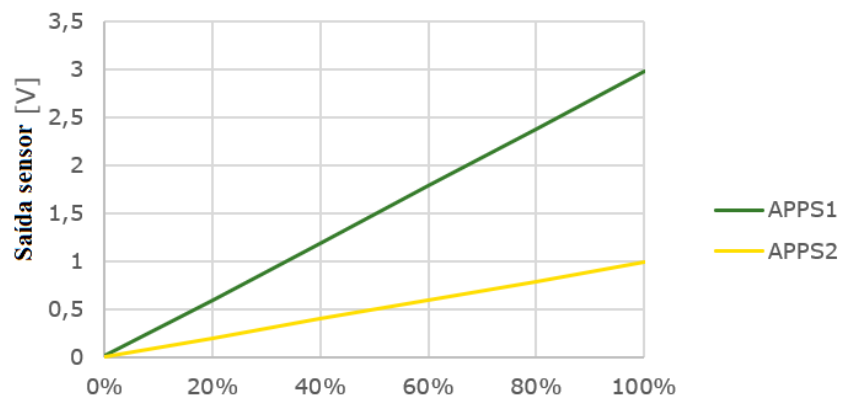
Figura 23 – Sensor de posição TPS.



Fonte: Elaborado pelo autor.

Visando uma segurança adicional a competição obriga a utilização de dois sensores diferentes, cada um com uma função de transferência distinta, sendo aceita uma diferença de inclinação ou *offset* entre as curvas. Essa distinção foi realizada pela equipe via hardware, utilizando a ECU do diferencial para tal feito. Na curva teórica da figura 24 percebe-se a diferença de inclinação escolhida para as duas funções de transferência. Neste caso, a saída do sensor 1 é três vezes maior que a saída do sensor 2.

Figura 24 – Funções de transferência dos sensores APPS.

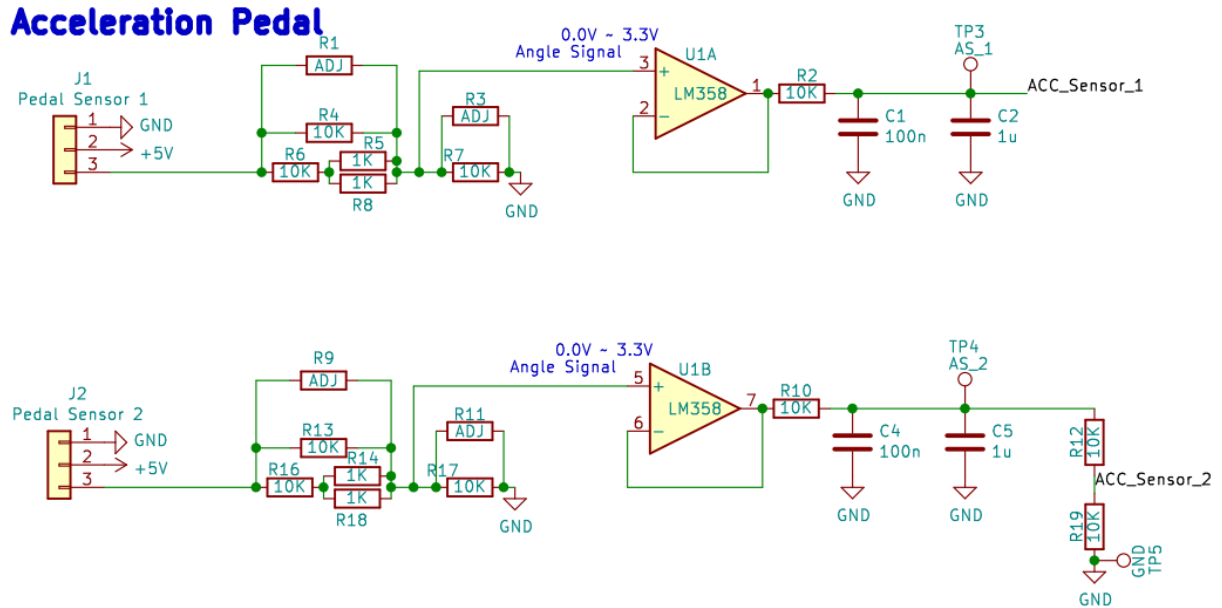


Fonte: Elaborado pelo autor.

Apesar do requisito da competição exigir funções de transferência distintas, outro requisito existe para garantir a redundância deste sistema. Um erro de implausibilidade é definido quando um desvio maior que 10% ocorre entre os sensores. Se essa diferença persistir por mais que 100ms, a alimentação do motor deve ser interrompida. Essa condição de comparação entre os sinais será implementada via software e a comunicação com o inversor será realizada via protocolo CAN. Vale ressaltar que para fazer a comparação entre os dois sinais o valor recebido do sensor dois é dividido por 3.

O condicionamento destes sinais está apresentado na figura 25.

Figura 25 – Condicionamento de sinal dos sensores APPS.



Fonte: Elaborado pelo autor.

Cada sinal passa por um amplificador operacional LM358 (alimentação não simétrica) operando como *buffer*, sendo que o ganho de cada circuito é ajustado pelos potenciômetros multivoltas R1, R3, R9 e R11. Percebe-se que a amplitude de saída do sinal *apps2* é 3 vezes menor que o *apps1*, devido ao divisor de tensão apresentado. Tal característica condiz com a curva apresentada na figura 24. A saída de ambos circuitos de condicionamento são lidas pelas entradas dos conversores analógico digitais do microcontrolador da placa ECU-diferencial. A mesma será instalada atrás do painel frontal do veículo, em um case impresso em 3D.

### 3.3 Sensor de ângulo da direção (SWAS - *Steering Wheel Angle Position*)

O sensor de ângulo de direção que será utilizado é um modelo comercial C68049XF do veículo *Trail Brazer* da *General Motors* (GM). O mesmo é ilustrado na figura 26.

Esse sensor opera com ângulos entre -150 a 150 graus, garantindo uma precisão de 0.1°. A escolha foi realizada devido ao custo benefício do mesmo, além do fato do volante da equipe ser projetado para atuar dentro dos limites de ângulo (-130 a 130 graus).

A saída do mesmo é digital, utilizando o protocolo CAN. Desta forma é necessário o uso de um *transceiver* CAN para a conexão entre o barramento físico e as entradas do microcontrolador. O módulo utilizado foi o circuito integrado MCP257FD/8FD, o qual

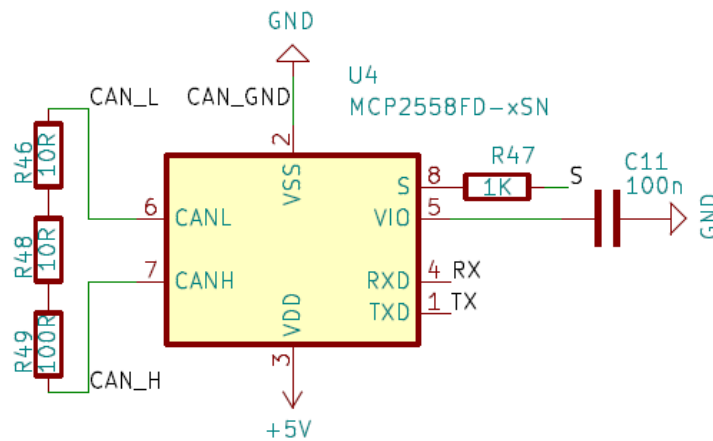
Figura 26 – Sensor de ângulo de direção.



Fonte: Adaptado de Eletrogate.

é utilizado em aplicações com CAN 2.0 e FDCAN na indústria automotiva, espacial e médica. O diagrama esquemático do mesmo é apresentado na figura 27.

Figura 27 – Diagrama esquemático MCP2557FD/8FD.



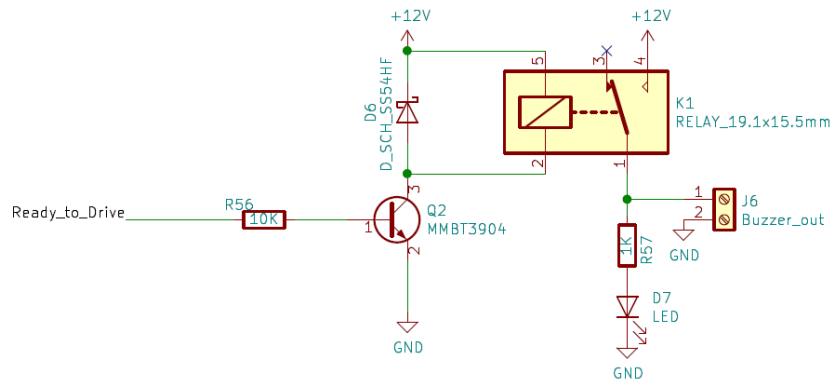
Fonte: Equipe e-Power UFRGS.

Assim, o sensor de ângulo é ligado ao *transceiver* através dos pinos 6 e 7, *CAN Low* e *CAN High* respectivamente. A saída serial é levada até o microcontrolador através dos pinos 4 e 1. O pino 8 (S) é uma função adicional do módulo, o *silent mode*, que será tratado posteriormente no capítulo do firmware. Vale ressaltar que um resistor equivalente de  $120\Omega$  foi adicionado ao final do barramento para fazer o casamento com a impedância característica do barramento.

### 3.4 Modo pronto para dirigir - *Ready to Drive Mode*

Dado o requisito das regras da FSAE e conforme descrito no capítulo 1.3.6 um modo pronto para dirigir deve ser implementado. O sistema desenvolvido para suprir essa demanda é ilustrado na figura 28.

Figura 28 – Circuito modo pronto para dirigir.



Fonte: Elaborado pelo autor.

Primeiramente, um botão de pressão localizado no painel frontal do carro é conectado a uma entrada digital do microcontrolador. Para evitar oscilações de leitura entre os níveis lógicos, um *debouncing* por software foi implementado. Quando tal botão e o pedal de freio forem pressionados simultaneamente, a saída digital do microcontrolador (identificado como *Ready to Drive*) assume um nível lógico um por cerca de 2 segundos, energizando a bobina do relé e consequentemente fechando o respectivo contato e acionando a sirene. O código que implementará tal solução será descrito no próximo capítulo.

O *buzzer* que será utilizado é uma sirene, a qual possui uma tensão nominal de 3 a 24V, corrente de operação de 12mA e pressão sonora de 90dB a 10cm do sensor. O mesmo foi escolhido para atender ao requisito da competição de garantir uma pressão mínima de 80dBA do lado de fora do veículo. Como tal intensidade sonora pode prejudicar a audição do piloto é previsto a confecção de uma proteção sonora feita por uma impressora 3D. A verificação dos níveis de intensidade sonora do *buzzer* será posteriormente testado pela equipe, utilizando um decibelímetro.

### 3.5 Instrumentação do pedal de freio

A instrumentação do pedal de freio será realizada a partir da utilização de dois sensores de pressão da linha de freio. O mesmo foi empregado tanto para detectar quando o pedal for pressionado, no caso do sistema de luz de freio, quanto para mensurar a pressão

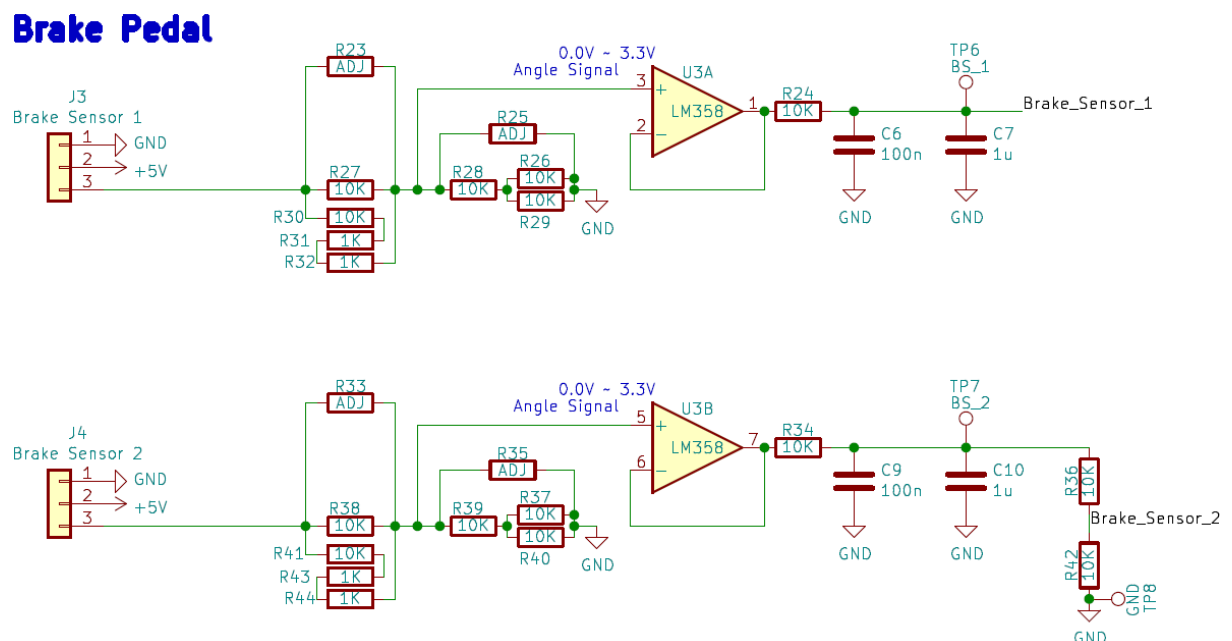


exercida pelo piloto (variável utilizada por outro sistema de segurança do carro, além de facilitar futuras validações do sistema hidráulico).

O sensor que foi escolhido mede pressões de 0 a 10 bar e é análogo ao ilustrado na figura 11. Tal valor fora escolhido devido ao fato do regulamento impor que o pedal de freio suporte uma força de 2000N, sem que ocorra falha no sistema de frenagem. Considerando o tamanho do pedal, a pressão máxima exercida pelo piloto ficaria em torno de 7 bar. A tensão de alimentação do sensor é de 8 a 30V e o *range* de saída varia de 0 a 5V.

O sistema a ser desenvolvido é destacado na figura 30. O mesmo é análogo ao circuito do pedal do acelerador. Isto é, possui um condicionador de sinal para cada sensor de pressão, de forma a garantir a redundância dos sinais. Esse circuito está localizado na placa da ECU do diferencial. Quando o firmware detectar que o pedal foi pressionado e a aquisição de pressão passou nos testes de integridade, um sinal de nível lógico um é enviado até a placa do circuito da luz de freio.

Figura 29 – Circuito condicionamento de sinal sensores de pressão de fluido de freio.

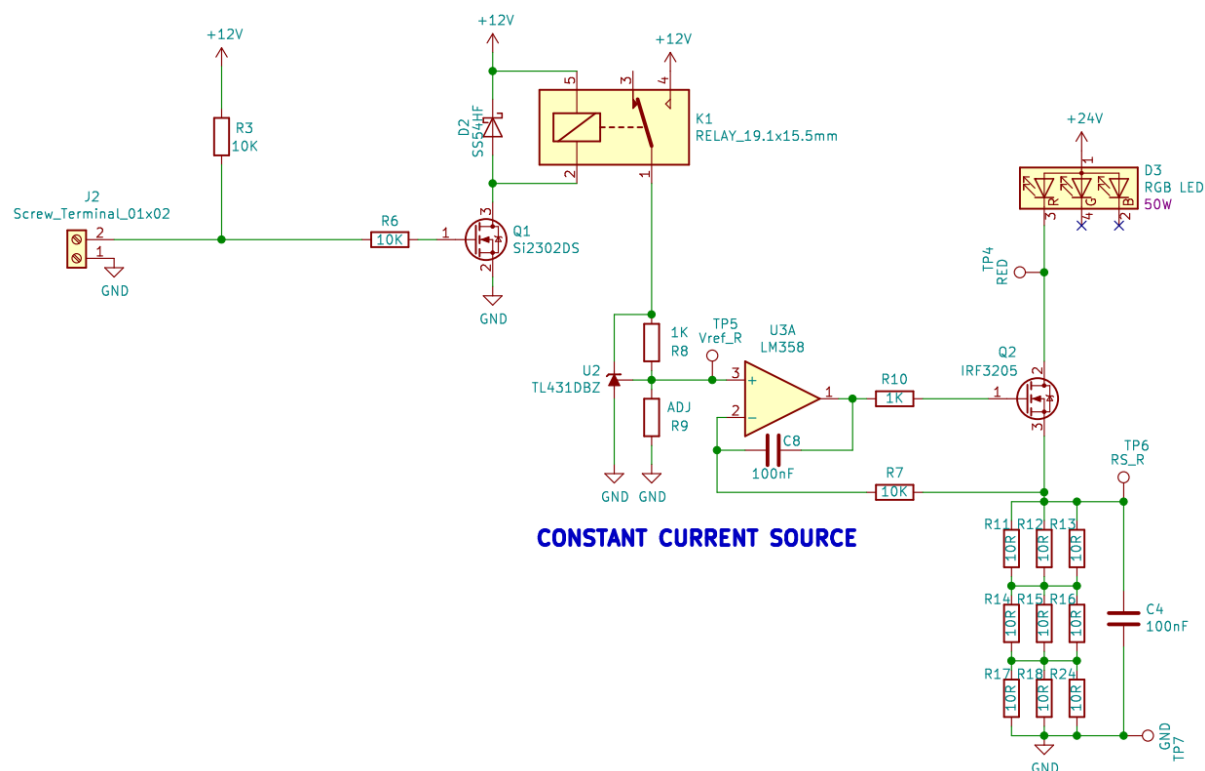


Fonte: Elaborado pelo autor.

O circuito de acionamento da luz de freio é apresentado na figura 30. Quando o pedal de freio for pressionado o contato do relé alimenta a fonte de corrente formada pelo amplificador operacional LM358 e o transistor NMOS IRF3205, sendo controlada pelo regulador de tensão TL431DBZ. Esse fato permite alterar o brilho do LED RGB de 50 W (figura 31) de forma que o mesmo seja visível sob luz intensa solar, requisito imposto pela competição. Vale ressaltar que o resistor R3 foi adicionado de forma que se

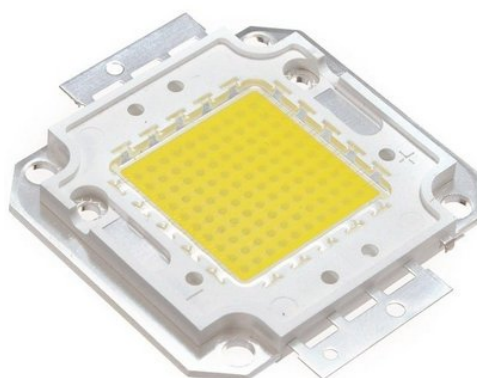
ocorrer um rompimento no sinal de entrada, a luz vai se manter sempre acesa, forma visual de detecção de eventuais falhas no sinal. O capacitor C8 foi utilizado para garantir a estabilidade do sistema.

Figura 30 – Circuito acionamento luz de freio.



Fonte: Elaborado pelo autor.

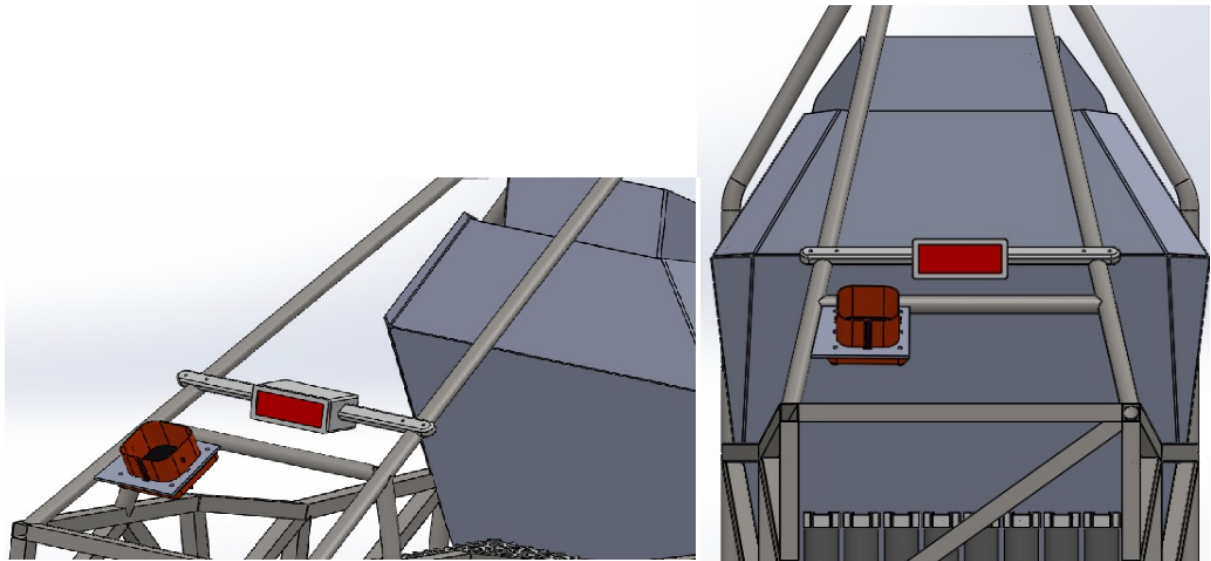
Figura 31 – LED RGB 50W.



Fonte: Elaborado pelo autor.

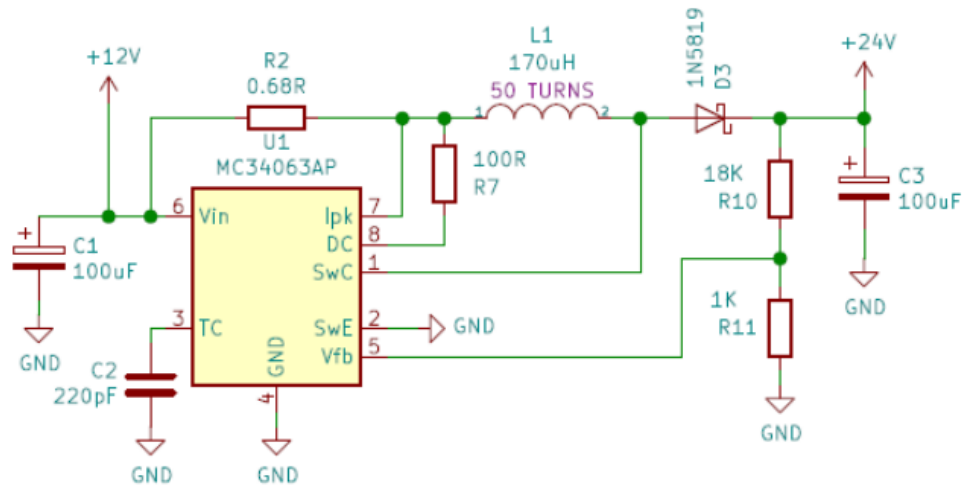
A junção do dissipador e do difusor (utilizado para direcionar o fluxo luminoso produzido pelo LED) será realizada a partir de uma caixa produzida por uma impressora 3D e a fixação será feita na parte traseira do chassi do veículo, de forma a deixar a luz o mais visível possível (caixa ilustrada na figura 32). Dentro dessa caixa será previsto um anteparo de alumínio para dissipação de calor do LED.

Figura 32 – Caixa da luz de freio e sua fixação no chassi do veículo.



Fonte: Elaborado pelo autor

Devido ao fato do relé não atuar diante de oscilações rápidas do sinal do interruptor de freio, não é necessária a utilização de um *debouncing* via hardware. A alimentação dos LED's é de 24V, para alcançar tal feito a equipe implementou um conversor DC/DC boost para elevar e regular a tensão da bateria de baixa tensão. Tal circuito é ilustrado na figura 33.

Figura 33 – Conversor DC/DC *Boost* para acionamento dos LED's do TSAL.

Fonte: Elaborado pelo autor.

A escolha dos componentes foi realizada segundo as fórmulas presentes no datasheet do circuito integrado, sendo que o indutor foi construído pela própria equipe. O mesmo irá integrar a placa de testes e a definitiva do pedal de freio.

### 3.6 Sistema de Segurança

A alimentação dos circuitos de segurança de baixa tensão será realizada a partir de quatro células em série e uma em paralelo, obtendo uma tensão de 14,8V e uma capacidade de fornecimento de 2500 mAh. A mesma será aterrada ao chassi do veículo (constituído de material com baixa resistência elétrica) pelo seu terminal negativo.

Devido ao fato desse circuito de segurança ficar em série com as bobinas dos dois relés da bateria, qualquer falha provoca a abertura dos contatos dos mesmos, interrompendo a alimentação com o inversor e consequentemente com os motores do carro. Esses respectivos relés de alta tensão, serão instalados dentro da caixa das baterias.

Os demais sistemas ilustrados na figura 17, dentro do container e relacionados com o sistema de tração, foram implementados pela equipe.

#### 3.6.1 Interruptor de Inércia

O sensor de inércia (ilustrado como *Inertia Switch* na figura 17) que será utilizado é o *Resetable Crash Sensor* da *Sensata Technologies* (figura 34). Os mesmos são uma solução de baixo custo para os requisitos envolvendo segurança de veículos.

Figura 34 – Sensor de Inércia.

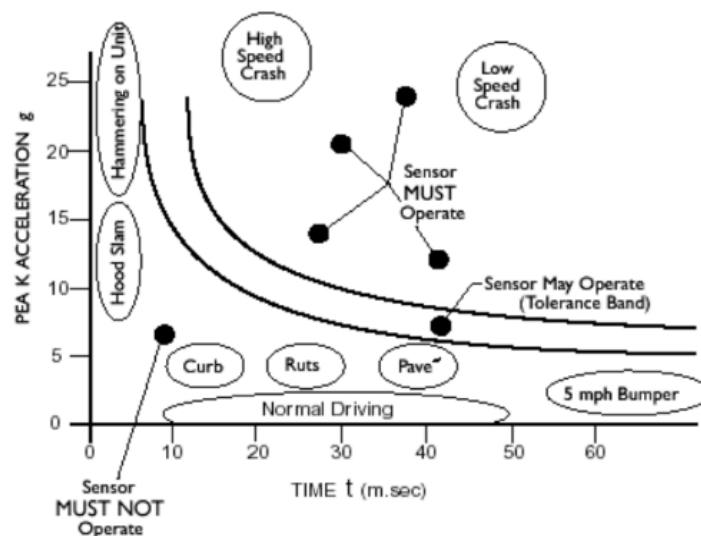


Fonte: *Sensata Technologies*.

A performance típica do mesmo pode ser visualizada a partir de um gráfico contendo a resposta do sensor em função da aceleração submetida e a sua respectiva duração (figura 35). Ou seja, considerando, por exemplo, que o carro sofra uma aceleração de 10 vezes a aceleração da gravidade por 10ms, a mesma não é suficiente para a atuação do sensor, todavia, se esse tempo fosse em torno de 50ms o sensor atuaria.

Segundo as regras da competição, o mesmo deve atuar para desacelerações de 8 a 10g, considerando tempos de duração dentro da margem de acionamento do dispositivo. Diante disso, o mesmo será fixado dentro do cockpit do veículo, de forma receber integralmente as variações de velocidade do carro durante a competição.

Figura 35 – Performance típica do sensor de inércia.

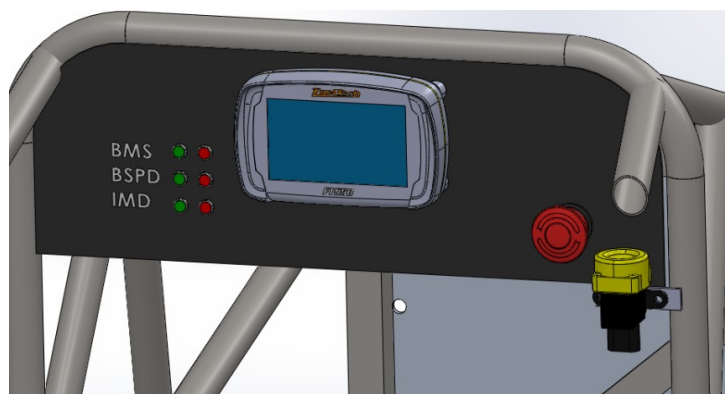


Fonte: *Sensata Technologies*.

Se o dispositivo for atuado, o circuito de desligamento do carro deve ser desacionado, assim os terminais do sensor serão ligados em série com esse circuito.

Conforme regra da competição, o teste sugerido para o sensor de inércia consistirá em desconectar o mesmo do carro e agitá-lo, simulando uma desaceleração do veículo e verificando assim se o circuito de desligamento atua.

Figura 36 – Performance típica do sensor de inércia.



Fonte: *Sensata Technologies*.

O sensor ficará localizado junto ao painel frontal do veículo (figura 36).

### 3.6.2 Sistema de checagem de plausibilidade do freio (BSPD)

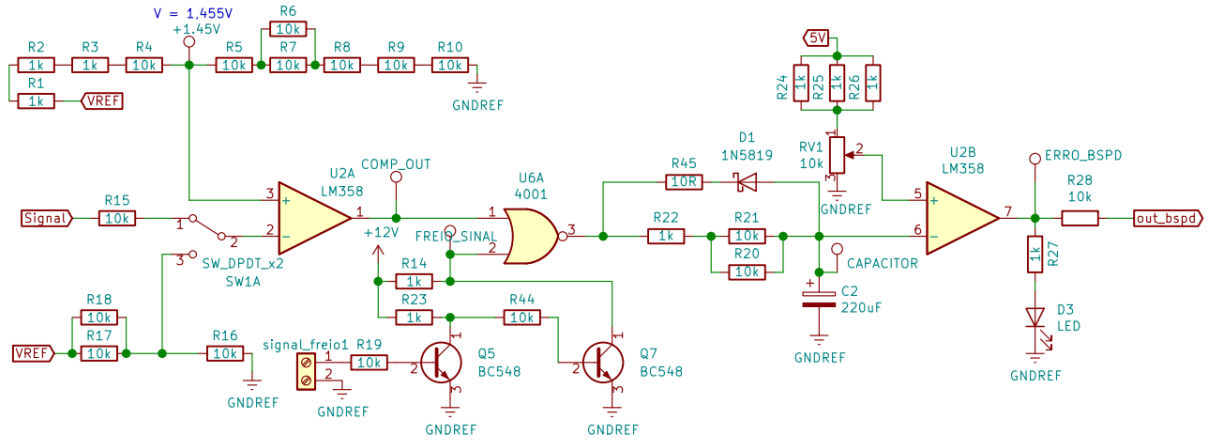
Como descrito nos capítulos anteriores esse circuito tem por objetivo o monitoramento da corrente que vai da bateria até o inversor e o acionamento do pedal de freio. O sistema proposto para atender tais requisitos da competição é ilustrado na figura 37. Vale ressaltar que a placa de circuito impresso desse circuito ficará localizado em um caixa junto com as outras placas, fixada em cima do acumulador.

Para a medição da corrente, será utilizado o circuito integrado ACS718. O mesmo é um sensor de efeito Hall linear de baixo ruído que, de forma geral, é um transdutor constituído por um caminho de cobre que sob aplicação de uma corrente externa gera um campo magnético, o qual acaba induzindo certa tensão na saída. Devido ao efeito físico, o circuito de alta tensão ficará isolado galvanicamente do circuito de baixa tensão.

A sensibilidade do sensor é de 100 mV/A, com um erro total de 1,5% a 25 °C e uma faixa de operação de -30A a 30A.

A partir do sensor de corrente ACS718, tem-se uma diferença de potencial proporcional a corrente contínua que passa entre a saída da bateria até a entrada do inversor de frequências. Segundo as normas da FSAE, essa corrente não deve provocar uma potência

Figura 37 – Diagrama sistema de checagem de plausibilidade do freio.



Fonte: Elaborado pelo autor.

maior que 5kW por 0,5 segundos, enquanto o pedal do freio estiver pressionado. Considerando a tensão nominal do sistema de tração como 355,2 V, obtém-se uma corrente máxima de 14,08 A e consequentemente uma tensão de 1,408 V na saída do sensor. Essa saída é identificada como *Signal* na figura 37. A mesma é utilizada junto com o comparador LM358 U2A, de forma a identificar quando o sinal ultrapassará o limiar mencionado. Se essa última condição for verdadeira e o pedal de freio for acionado (nível lógico 0 na entrada identificada como *Signal freio* na figura), Q5 estará em corte e Q7 saturado, condição que faz a saída da porta NOR assumir um nível lógico 1. Assim, o capacitor C2 começa a carregar.

O filtro RC formado pelo resistor R22, R21, R20 e o capacitor C2 tem por objetivo, atuando junto com outro comparador, garantir que a condição de 0,5 segundos seja satisfeita.

Dado que a resposta ao impulso ( $h_C(t)$ ) do circuito formado por um resistor e capacitor, em função da resistência (R) e capacitância (C) é dada por:

$$h_C(t) = (1/RC)e^{-t/RC}u(t) = (1/\tau)e^{-t/\tau}u(t) \quad (3.1)$$

Onde  $u(t)$  é a função de *Heaviside* e  $\tau$  é definida como a constante de tempo. Aplicando os valores da figura 37 obtém-se um  $\tau$  de 2.68 segundos, após  $5\tau$  o capacitor está praticamente carregado (99.3 % da carga). Dessa forma, se a condição de implausibilidade persistir por mais de 0,5 segundos, a tensão sob o capacitor será maior do que a gerada com o divisor de tensão usando o potenciômetro multi-voltas RV1 (de forma a garantir um ajuste fino experimental), ou seja, o segundo comparador passa a assumir um nível lógico

"0". Essa saída vai para a placa responsável pela interface de potência com o circuito de desligamento, a placa denominada *set-reset* (detalhada no capítulo posterior).

Se alguma das condições de implausibilidade for violada, potência fornecida pela bateria for menor que 5 kW ou o pedal do freio não pressionado, a tensão sob o capacitor é descarregada pelo diodo de sinal. Fato que faz com que a saída do segundo comparador assuma um nível lógico "1".

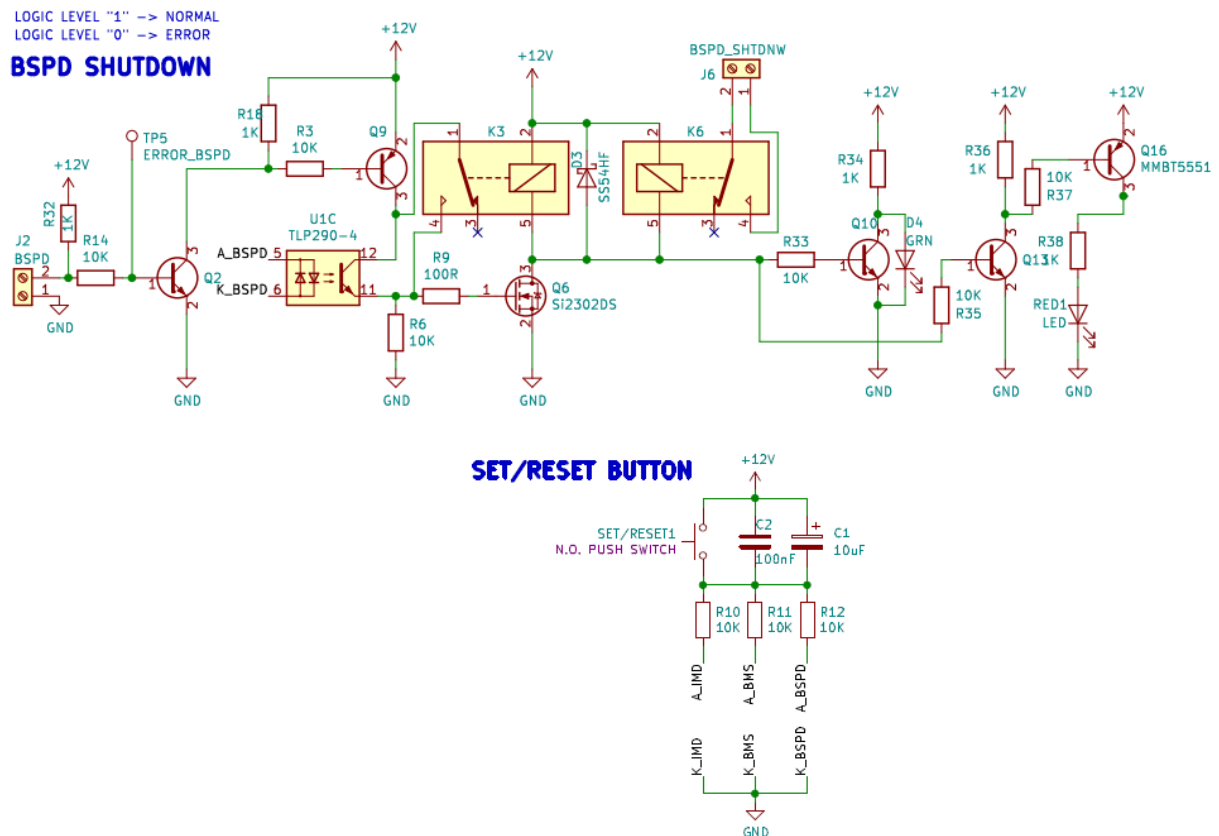
De forma a testar a saída do sensor de corrente, sem a necessidade de utilizar uma fonte de corrente, foi incluída uma chave seletora na entrada no primeiro comparador. Assim, é possível demonstrar o funcionamento do circuito na competição utilizando um simples divisor de tensão.



### 3.6.3 Circuito de Set/Reset do sistema

Em caso de falha nos circuitos do BMS, BSPD ou IMD um botão de reset na parte traseira do veículo deverá ser pressionado. O circuito responsável por tal feito, para o caso do BSPD é ilustrado na figura 38. Os circuitos para o BMS e o IMD são análogos. A placa de circuito impresso do circuito ficará localizada junto com as demais de baixa tensão.

Figura 38 – Circuito Set/Reset.



Fonte: Elaborado pelo autor.

Para cada um dos três circuitos anunciados, a condição de necessidade de acionamento do botão de reset é descrita como segue: Em caso de normalidade a saída dos circuitos possui um nível lógico "1", caso contrário, a respectiva saída assume um nível lógico "0". Três LED's indicativos vermelhos serão instalados no cockpit do carro, caso ocorra algum erro o respectivo LED ficará aceso.

A saída do circuito do BSPD, por exemplo, é conectada na entrada do circuito da figura 38. Em condições normais de funcionamento, a mesma possui nível lógico um. Nesse caso, quando o circuito de baixa tensão é ligado, os capacitores (C1 e C2) operam como um curto, fato que tem por consequência o acionamento do optoacoplador e do relé (acionado pelo transistor Q4). Assim, o circuito de desligamento é fechado (conector BSPD

SHTDNW) e a bobina do relé K3 passa a ser alimentada devido ao primeiro contato do mesmo. Quando uma falha é detectada a entrada passa a assumir nível lógico zero, fazendo com que os transistors Q2, Q9 e Q6 operem na região de corte. Tal efeito ocasiona o acionamento de um LED vermelho no cockpit, indicando a respectiva falha e a abertura do circuito de desligamento. Se o sistema voltar a operar normalmente, isto é, a entrada voltar a assumir um nível lógico "1", o circuito de desligamento somente é fechado quando o botão de set/reset for pressionado.

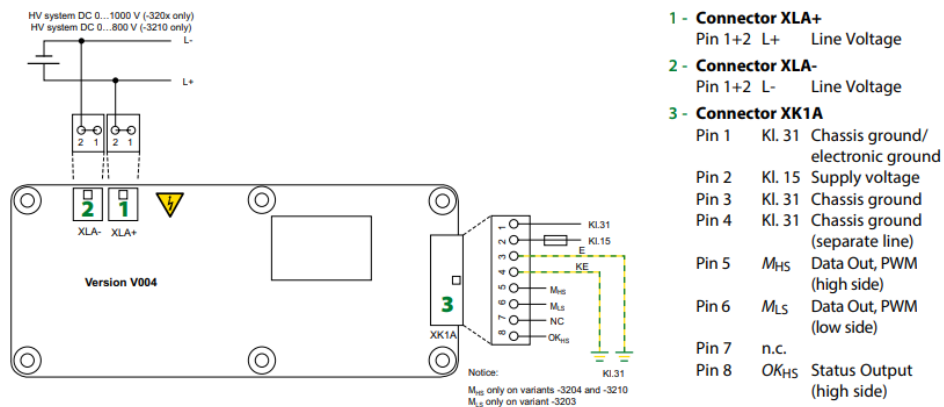
### 3.6.4 Sistema de monitoramento do nível de isolamento

O sensor a ser utilizado para monitorar a impedância entre o sistema de alta tensão e o chassi do veículo é o Bender A-ISOMETER. Nesse caso, se qualquer curto-circuito vier a ocorrer entre esses dois sistemas o circuito de desligamento deverá ser aberto.

Segundo os requisitos da competição, essa abertura deverá ocorrer quando uma impedância menor que  $500\Omega/V$  for detectada entre os respectivos sistemas, assim, utilizando a tensão máxima das baterias de 355,2 V obtêm-se que a impedância deverá ser maior que  $177,6k\Omega$ . Uma informação relevante é que o sensor atual da equipe é acionado para impedâncias inferiores a  $294k\Omega$ , ou seja, ainda pode ser utilizado para a tensão atual do veículo.

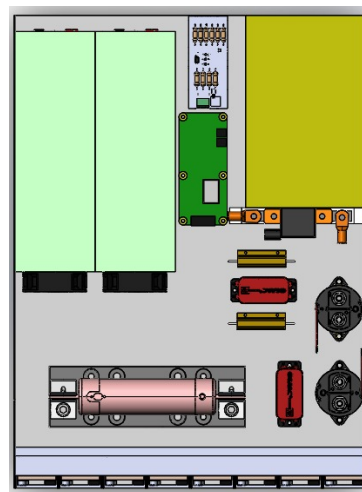
Como é necessária somente a detecção de curto-circuitos no sistema, só essa respectiva saída do sensor será utilizada (pino 8 conforme figura 39). Essa saída discreta será conectada a entrada do circuito do IMD localizado na placa do set/reset. Os pinos 3 e 4 do conector XK1A serão soldados no chassi e os pinos 1 e 2 do conector XLA+ serão conectados aos terminais positivo e negativo do sistema de tração. Este módulo será instalado dentro do andar superior da caixa do acumulador, conforme ilustrado na figura 40 (placa verde localizada no centro, os demais componentes são relacionados com a parte de *powertrain* da equipe).

Figura 39 – Diagrama do sensor Bender A-ISOMETER.



Fonte: Bender.

Figura 40 – Caixa do acumulador visão superior.



Fonte: Elaborado pelo autor.

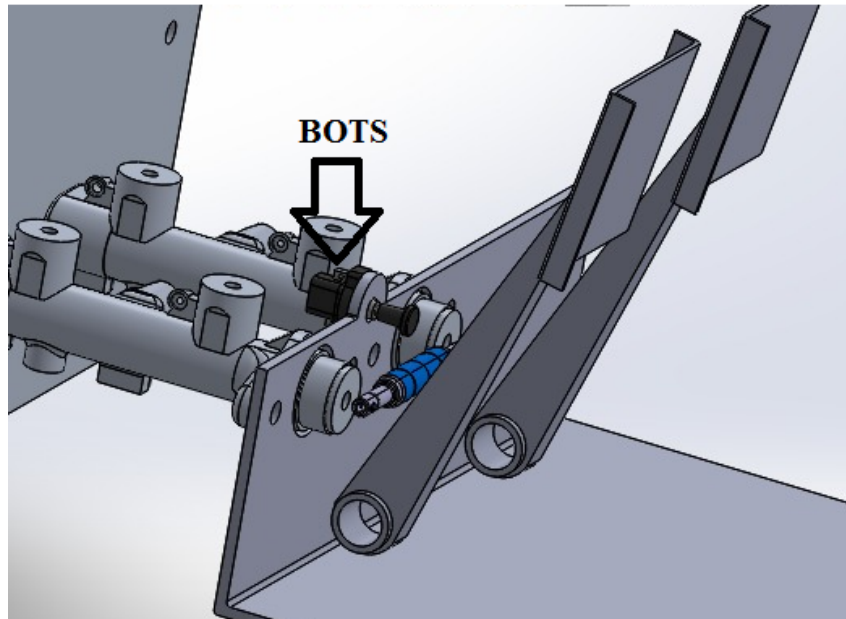
De forma a testar o circuito posteriormente, como ocorre na competição, é previsto a execução de testes em bancada, inserindo uma resistência da metade do valor calculado anteriormente ( $88.8k\Omega$ ) entre o sistema de tração e o chassi do veículo, de forma a verificar se a saída do IMD muda. Posteriormente, vários valores de resistência serão testados de forma a verificar o chaveamento do sensor.

### 3.6.5 Sistema de ultrapassagem do alcance do pedal de freio (BOTS)

O sistema de segurança utilizado para desligar o sistema de tração em caso de alguma falha nos freios do veículo vai ser constituído por um botão de pressão com trava,

instalado atrás do pedal de freio. Os terminais do botão ficarão em série com o circuito de desligamento, de forma a abrir os relés da bateria em caso do pedal ultrapassar os limites de atuação. Devido a localização do mesmo, após ser pressionado, somente um membro da equipe fora do carro poderá resetar o sistema. O botão será o mesmo que o ilustrado na figura 13.

Figura 41 – Fixação BOTS.



Fonte: Elaborado pelo autor.

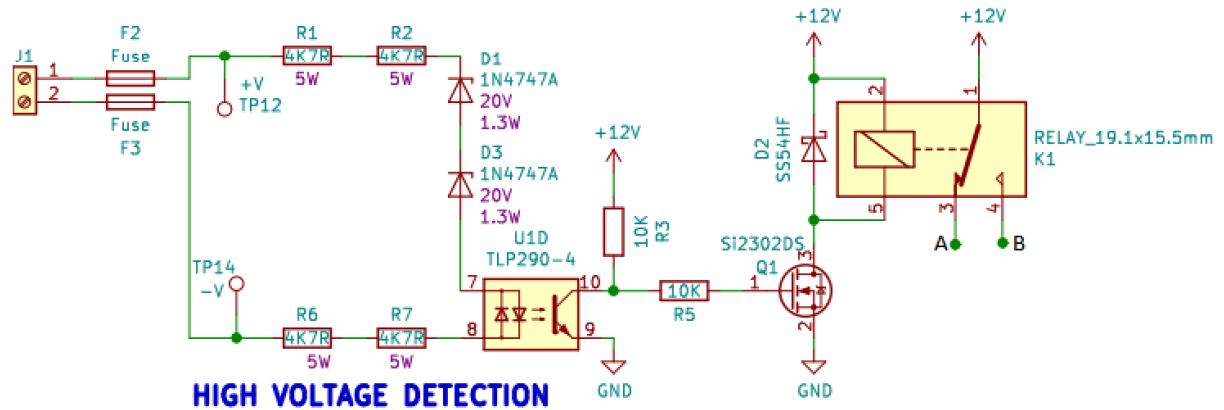
Na figura 41 pode-se observar como o sensor será fixado o botão atrás dos pedais.

### 3.7 Luz indicativa do sistema de tração (TSAL)

O circuito proposto para a implementação da luz indicativa do sistema de tração é ilustrado na figura 42. Como descrito nos capítulos anteriores, esse circuito é responsável pela indicação luminosa do estado do sistema de tração do veículo.

Os terminais da bateria de alta tensão serão conectados aos conectores 1 e 2 na esquerda do circuito da figura 42. Primeiramente, quando o sistema de tração se encontra em alta tensão (maior que 60VDC), a disposição dos diodos zeners e resistores é tal que tem-se uma corrente suficiente para o acionamento do optoacoplador. Tal efeito tem por consequência deixar o transistor Q1 em corte e o relé não é acionado. Caso contrário (tensão na entrada menor que 60 VDC), o relé é acionado.

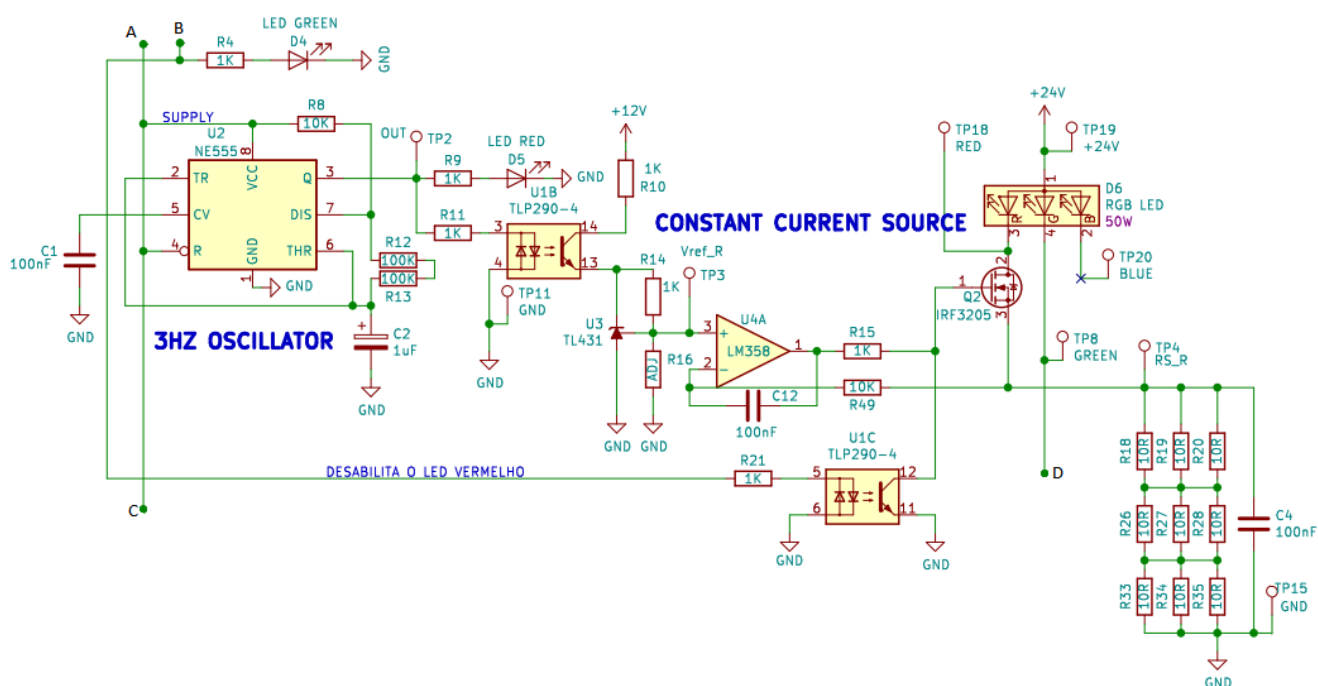
Figura 42 – TSAL - Detecção de alta tensão.



Fonte: Elaborado pelo autor.

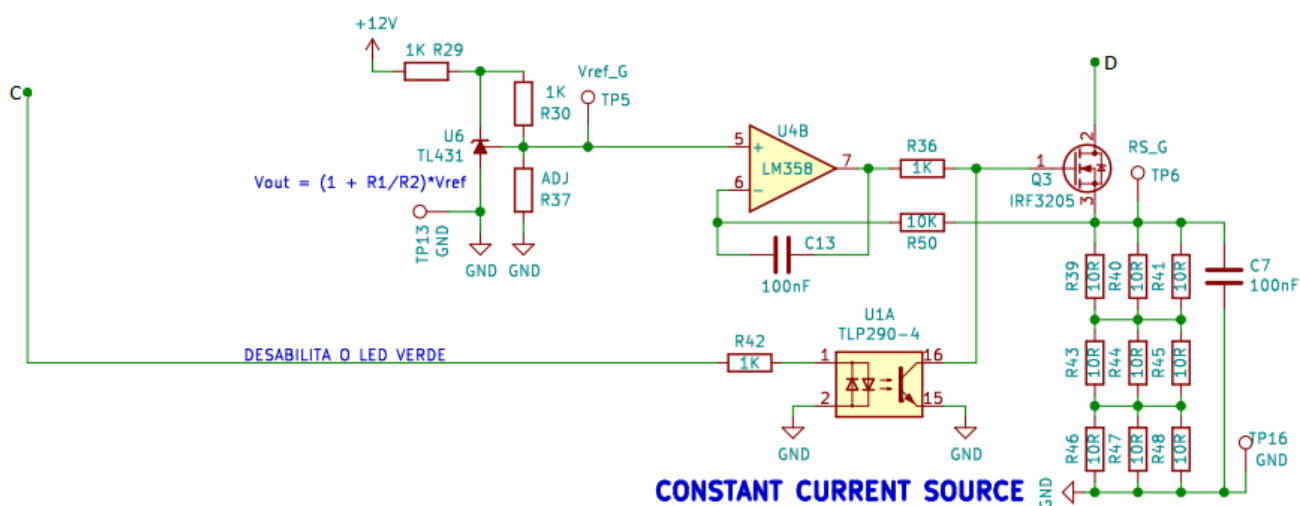
Supondo que o somente o sistema de baixa tensão esteja acionado no carro (sem tensão no barramento de alta tensão), o relé energiza o contato 4 (normalmente aberto), acionando o optoacoplador U1C da figura 43 e fazendo com que o transistor NMOS Q2 entre em corte, fato que tem por consequência o desligamento do setor vermelho do LED RGB. Caso a tensão no sistema de tração seja maior que 60V, o contato 3 do relé é energizado, Q2 não se encontra mais em corte e o LED RGB vermelho é ativado pelo CI 555. Esse último está configurado como monestável, isto é, gera uma saída com um clock de 3Hz para acionar o respectivo LED. O circuito de acionamento do LED verde (figura 44) é análogo ao descrito previamente, porém o mesmo não deverá piscar.

Figura 43 – TSAL - Acionamento RGB vermelho.



Fonte: Elaborado pelo autor.

Figura 44 – TSAL - Acionamento RGB verde.



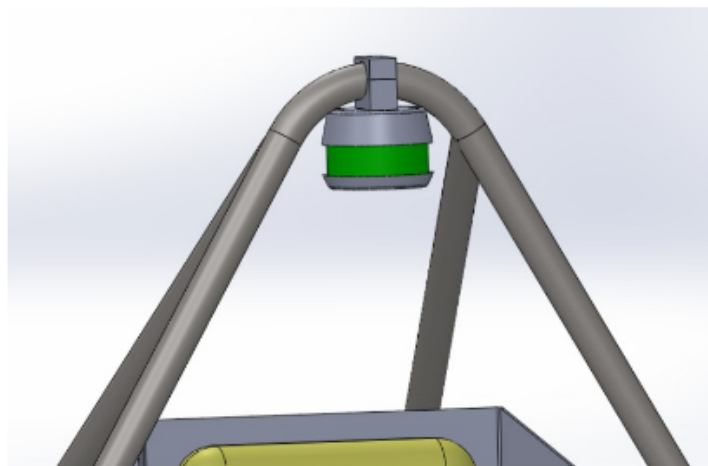
Fonte: Elaborado pelo autor.

Como cada cor do LED RGB necessita de correntes distintas para adquirir a mesma intensidade luminosa, e visando uma futura calibração, fontes de correntes controladas pela referência de tensão TL431 foram adicionadas ao circuito do TSAL.

A alimentação dos LED's é de 24V, para alcançar tal feito a equipe implementou um conversor DC/DC boost para elevar e regular a tensão da bateria de baixa tensão. Tal circuito é ilustrado na figura 33. O mesmo irá integrar a placa de testes e a definitiva do TSAL.

Além disso, outro requisito necessário é de que o TSAL só venha a informar sobre o estado do sistema se o circuito de baixa tensão estiver funcionando, ou seja, se o circuito de desligamento for fechado. Como a alimentação dos LED's do TSAL é pelo sistema de baixa tensão, isto é, dependendo do estado do mesmo, essa condição é satisfeita.

Figura 45 – Instalação do case do TSAL no carro.



Fonte: Elaborado pelo autor.

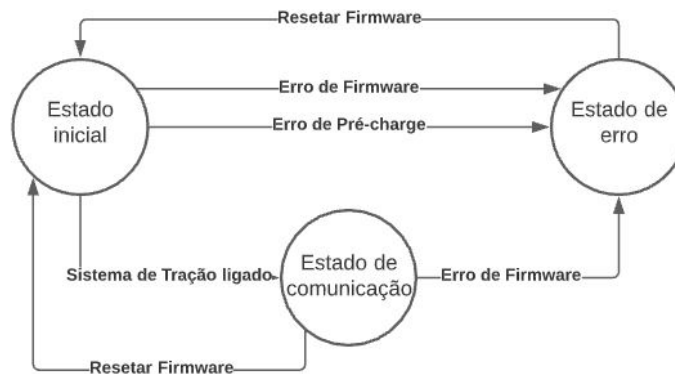
Na figura 45 é ilustrado como o case do TSAL será fixado no veículo.

## 4 Desenvolvimento do Firmware - Diferencial

O presente capítulo visa descrever desenvolvimento do firmware de uma das ECUs do carro, a ECU do diferencial. Os demais firmwares estão em desenvolvimento no presente momento.

De forma a organizar melhor o código foi feito o uso do conceito de máquinas de estado, as quais impõe que o programa se encontre em um dos finitos estados possíveis. A mesma é ilustrada na figura 46.

Figura 46 – Máquina de estados - firmware ECU diferencial.



Fonte: Elaborado pelo autor.

Cada estado será descrito nos sub-capítulos a seguir, sendo que o código completo utilizado encontra-se no apêndice A.

### 4.0.1 Estado Inicial - *Init State*

O código se encontra no estado inicial quando o microcontrolador for alimentado. Isso ocorre quando a chave de baixa tensão do veículo é acionada. Segundo as regras da competição, para que o inversor comece a se comunicar com os motores certos requisitos precisam ser cumpridos.

A condição para que a placa comece a se comunicar com o inversor do veículo é a entrada no modo pronto para dirigir. Assim, se o botão de *ready to drive* do painel for pressionado junto com o pedal de freio e o sistema de tração estiver acionado, essa condição é satisfeita. Para a verificação do correto acionamento do botão foi utilizado um *debouncing* via software, checando se após 50 ms o botão permanece acionado.



A certificação que sistema de tração está acionado, ou seja, que o barramento de alta tensão do carro está ligado é recebida via protocolo CAN. Essa mensagem de certificação é enviada via CAN pelo circuito da ECU principal, a qual se comunica com o circuito de pré-carregamento do carro. Após essas condições o sistema vai para o estado de comunicação.

#### 4.0.2 Estado de Comunicação - *Communication State*

Nesse estado os conversores ADC de 12 bits começam a ler os valores dos sensores do pedal do acelerador e o sensor de pressão da linha de freio com uma frequência de amostragem de 1,41 MHz. É interessante ressaltar que foi feito uma média de 10 medidas para estimar a posição de cada pedal. Como cada medida possui um *range* de valores possíveis, pode-se concluir se houve algum curto ou circuito aberto. Tal condição é tratada como uma implausibilidade, e se persistir por mais de 100 ms o código vai para o estado de erro.

Para o cálculo de cada rotação foi utilizado o modelo de Ackermann-Jeantand apresentado na equação 2.1, utilizando uma distância entre eixos (D) de 1190 mm e uma distância entre rodas de 1700 mm. Essas rotações serão então enviadas via CAN para a ECU do inversor.

O regulamento também impõe que a equipe certifique a respeito da integridade do sinal do APPS, desde a saída analógica do sensor até a transmissão digital até o inversor. Primeiramente foi implementado uma checagem se o sinal analógico se encontra dentro dos limiares considerados. Para a checagem de integridade dos dados digitais foi utilizada a soma de verificação CRC (*Cyclic Redundancy Check*) de 17 bits do CAN, a qual inclui no final de cada pacote de dados um valor de verificação baseado no resto de divisão polinomial do seu conteúdo, se houver distinções entre os valores transmitidos e recebidos o firmware detectará o erro.

#### 4.0.3 Estado de Erro - *Error State*

Se alguma implausibilidade, erro de código ou falha na comunicação CAN ocorrer o sistema entra neste estado. Se a falha não ocorrer no protocolo de comunicação, uma mensagem de erro é prevista, de forma a alertar as demais ECUs conectadas no barramento a ignorarem os dados de rotação recebidos.

Independente do estado que o firmware estiver foi implementado um temporizador *Watchdog*, o mesmo dispara um reset ao sistema se o programa principal, devido a alguma condição de erro, deixar de fazer reset no temporizador, isto é, trata-se de um sistema emergencial para atestar que o sistema está funcionando normalmente. O *timeout* utilizado foi de 1 segundo, valor que é maior do que a maioria dos eventos do código.

## 5 Experimentos e Resultados

O presente capítulo visa ilustrar todos os experimentos necessários para confirmar o funcionamento dos circuitos descritos nos capítulos anteriores. Para cada circuito foi realizado uma simulação via software e para alguns foi possível a execução de experimentos em laboratório, de forma a confrontar esses dois resultados. Vale ressaltar que devido a situação atual, somente os circuitos mais complexos foram simulados e testados em laboratório.

Para a análise dos circuitos será utilizado o software LTSpice XVII. O mesmo foi escolhido pois além de ser de código aberto possui modelos matemáticos realísticos para a maioria dos componentes utilizados.

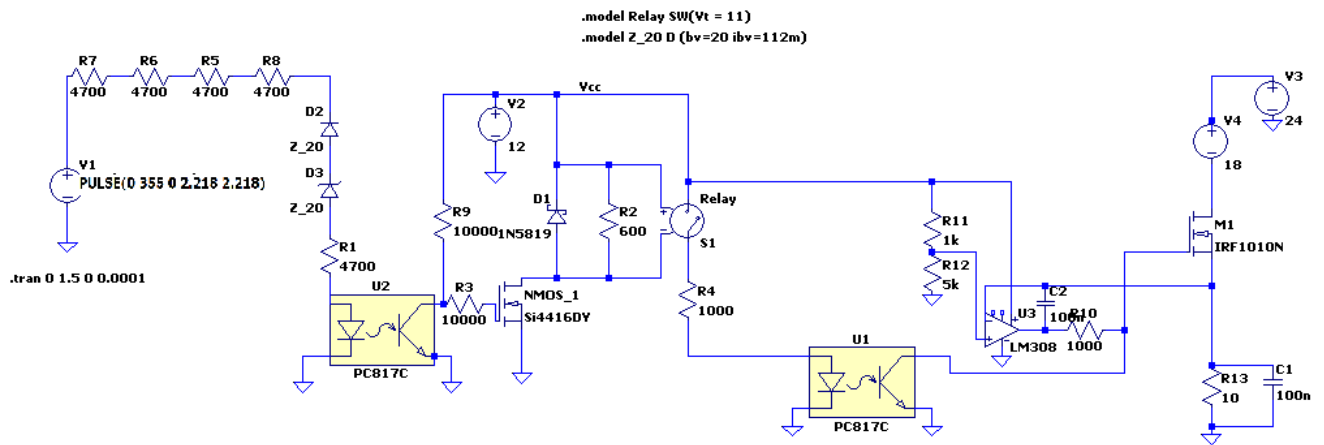
Para as atividades em laboratório optou-se primeiramente pelo teste dos circuitos em protoboard e posteriormente o uso de placas universais, de forma a identificar possíveis problemas antes da confecção das mesmas em circuito impresso.

### 5.1 Experimentos com a Luz Indicativa do Sistema de Tração (TSAL)

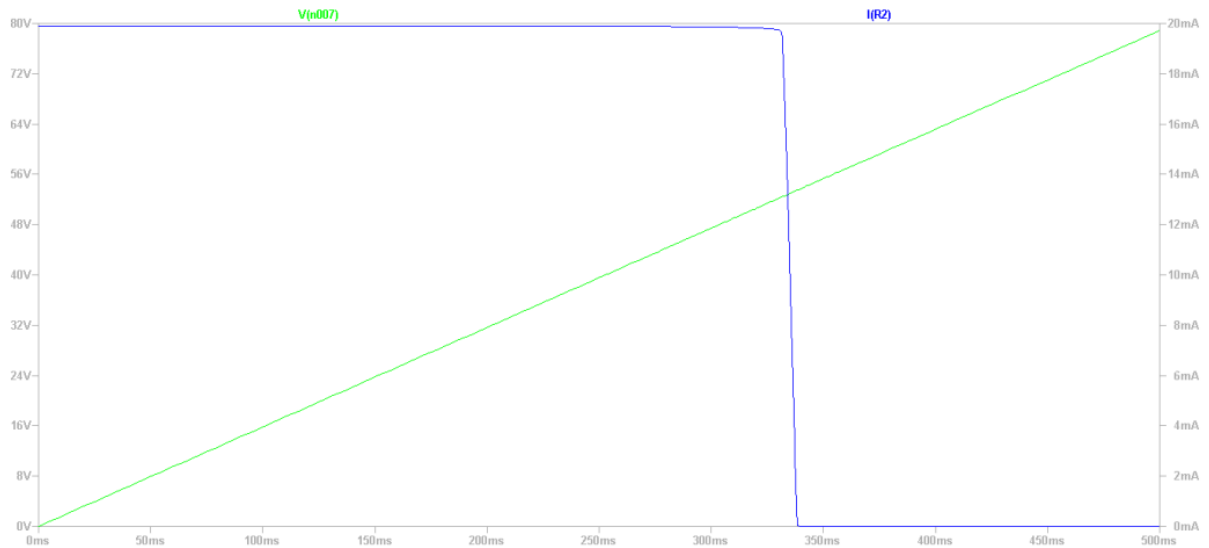
#### 5.1.1 Simulação

O circuito do TSAL, ilustrado na figura 47, foi simulado no modo transiente utilizando uma taxa de amostragem de 0.0001 segundos. Como nem sempre é possível a escolha de todos os componentes utilizados pelo software, similares foram empregados, de forma a aproximar a simulação com o experimento em bancada.

Na primeira simulação foi proposto verificar a tensão de chaveamento do optoacoplador, para isso a tensão de entrada foi aumentada linearmente, simulando um aumento da tensão das baterias. Para isso foi monitorada a corrente do resistor R2. Na figura 50, percebe-se a corrente na saída do optoacoplador surge com tensões a partir de 54 V. Isto é, é nessa tensão que ocorre o chaveamento do sistema, indicando que o barramento das baterias está em um estado de alta tensão.

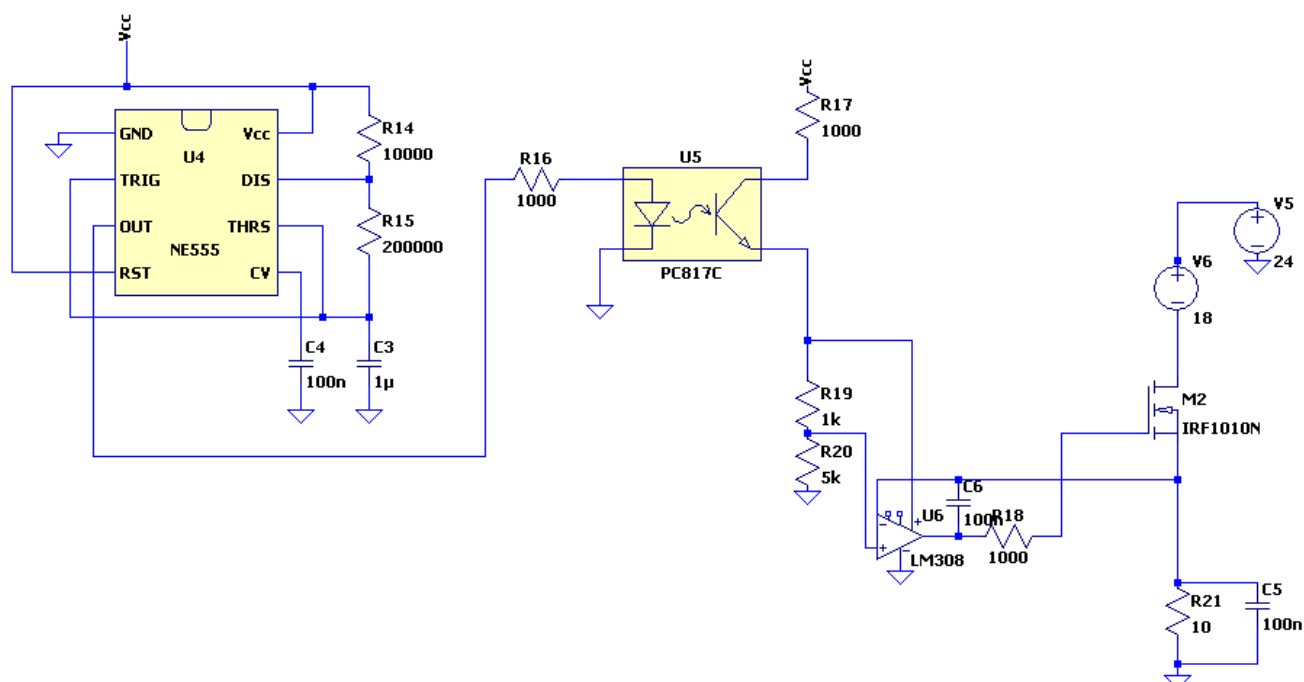
Figura 47 – Circuito do TSAL simulado no *LTSpice*

Fonte: Elaborado pelo autor.

Figura 48 – Corrente de saída do Optoacoplador ( $I(R2)$ ) em função da tensão de entrada (V).

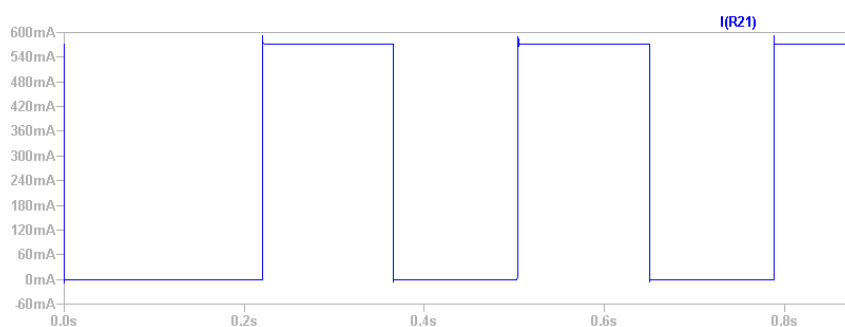
Fonte: Elaborado pelo autor.

A próxima simulação consiste em testar o funcionamento do contador 555 em conjunto com a fonte de corrente. O circuito simulado é apresentado na figura 49.

Figura 49 – Circuito de acionamento contador 555 simulado no *LTSpice*

Fonte: Elaborado pelo autor.

Figura 50 – Simulação temporização circuito 555



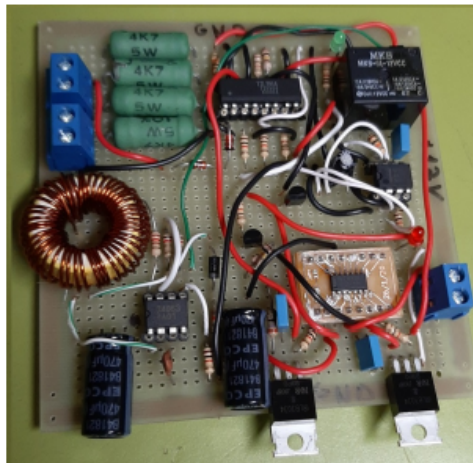
Fonte: Elaborado pelo autor.

Na figura 50 pode-se observar a corrente no dreno do transistor da saída com o formato previsto, com uma frequência de aproximadamente 3,50 Hz, fato que condiz com o regulamento (2 a 5 Hz).

### 5.1.2 Experimento em laboratório

A partir do esquema do circuito do TSAL foi realizado a confecção de uma placa de testes universal (figura 51) a qual foi testada em bancada com uma fonte de tensão. Aumentando a tensão de 0 até 100V foi possível verificar em qual valor o LED RGB mudava de verde para vermelho piscante.

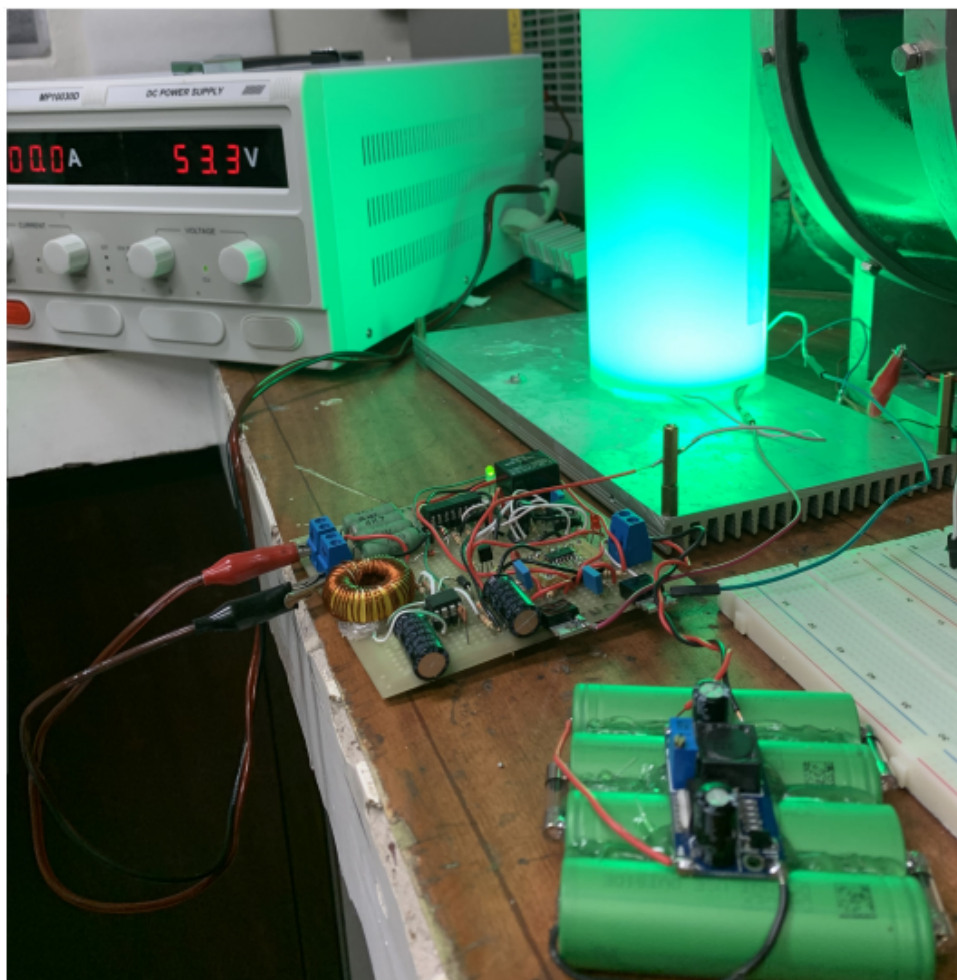
Figura 51 – Placa universal do circuito do TSAL.



Fonte: Elaborado pelo autor.

Na figura 52 é possível observar a montagem em bancada. Na prática a tensão de chaveamento encontrada foi de 57V. Tal diferença entre o simulado e o experimento pode ser justificada devido a falta de alguns componentes para a simulação e a tolerância dos componentes reais, como o caso resistores de potência (10 %).

Figura 52 – Testes em bancada da placa universal do TSAL.



Fonte: Elaborado pelo autor.

Como é feito o uso de componentes que dependem da temperatura, entre outros fatores, essa tensão poderia ultrapassar os 60 V no dia da competição, fato que justifica a escolha de um limiar menor.

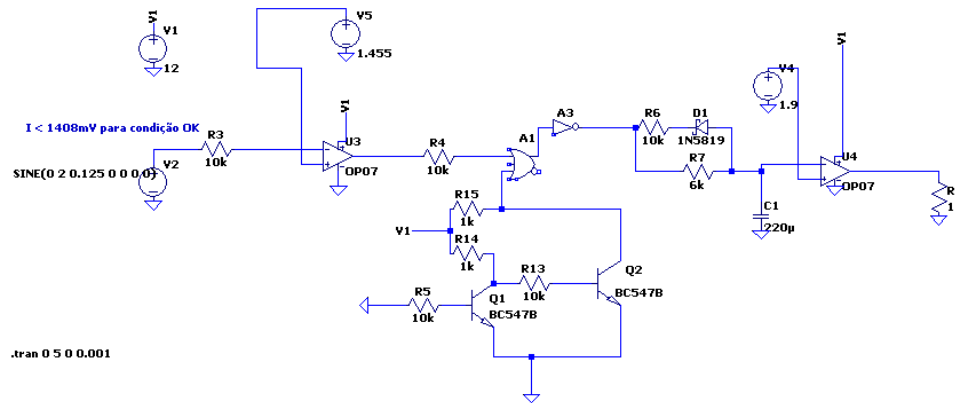
## 5.2 Experimentos com o Sistema de Checagem de Plausibilidade do freio (BSPD)

### 5.2.1 Simulação

A simulação do circuito do BSPD é ilustrada na figura 54. Para efeitos de simulação foi incluída uma fonte de tensão senoidal V2, simulando a saída do sensor de corrente devido ao aumento da corrente das baterias. O pedal de freio foi considerado sempre

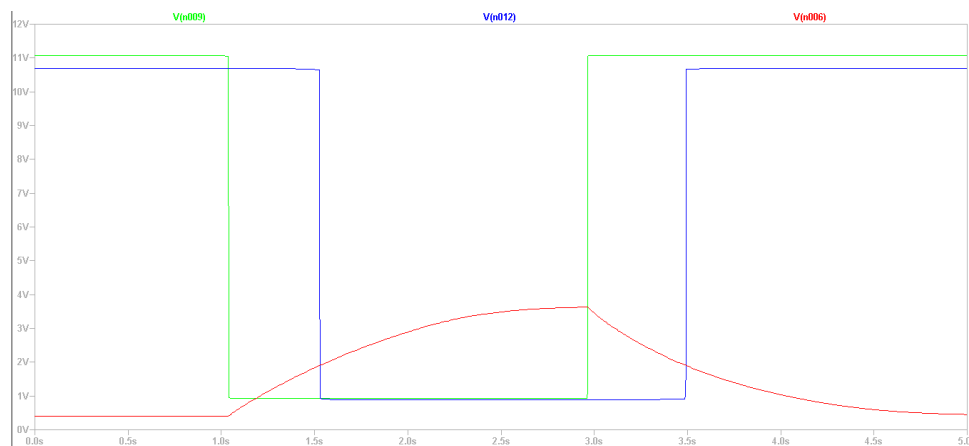
pressionado. Na figura 42 é possível verificar a saída do primeiro comparador (U3) em verde e do último (U4) em azul, além da tensão sob o capacitor em vermelho.

Figura 53 – Circuito do BSPD simulado no *LTSpice*.



Fonte: Elaborado pelo autor.

Figura 54 – Saída dos comparadores em função da tensão sob o capacitor.



Fonte: Elaborado pelo autor.

Percebe-se que com o aumento das correntes fornecidas aos dois inversores a saída de U3 varia do nível lógico "1" ao "0". Se tal condição permanecer por mais que 0.485 segundos (valor simulado), a saída de U4 assume um nível lógico "1", indicando uma falha. É possível verificar também a curva de carregamento e descarregamento do capacitor, essa última devido a corrente da bateria voltar a níveis normais de operação.

### 5.3 Experimentos com o firmware do diferencial

De forma a se certificar a respeito do funcionamento do firmware desenvolvido no presente trabalho foi realizado testes experimentais com o microcontrolador em uma *protoboard*, visando obter uma resposta rápida quanto ao funcionamento da ECU. Como nem todos os sensores estavam disponíveis para testes, optou-se pela utilização de divisores de tensão resistivos e potenciômetros, de forma a simular a resposta dos mesmos na ECU. O modo de execução do programa foi via depurador, adicionando pontos de interrupção ao longo do código para analisar passo a passo o funcionamento.

Primeiramente foi testado a interrupção gerada pelo pressionamento do botão de pressão, verificando se de fato o código checava as condições iniciais para entrar no modo pronto para dirigir. Se tudo estiver correto, foi verificada a mudança do estado inicial para o modo de comunicação da máquina de estados.

Com o firmware rodando no modo de comunicação, foi verificado se os ADCs estavam efetuando a leitura correta dos valores de sensor de ângulo e acelerador, resultando em uma medida da rotação de cada roda. Como o *transceiver* CAN não estava disponível para testes, não foi possível verificar o funcionamento do protocolo de comunicação CAN.

Por fim, foi testado o estado de erro a partir da simulação de valores das saídas dos sensores fora dos limiares, ou seja, supondo um curto ou circuito aberto. O firmware de fato entrou no modo de erro quando isso ocorria.

Assim, a partir de testes simples foi possível verificar o funcionamento do código desenvolvido.



## 6 Conclusão

Este trabalho propiciou um grande aprendizado tanto na área de Hardware, quanto na área de Software embarcado. O processo de desenvolvimento necessitou de intensa pesquisa em relação à eletrônica automotiva e suas interfaces com a área da mecânica, além dos requisitos impostos pela competição. Devido à complexidade do projeto eletrônico como um todo, somente parte dele foi desenvolvida no trabalho, como os elementos necessários para a obtenção das rotações do veículo, isto é, a ECU do diferencial. Módulos de identificação visual do status do sistema, como a luz de freio e o TSAL. Além dos sistemas de segurança do carro, como módulos de checagem de implausibilidades e sistemas mecânicos de atuação diante de emergências.

Verificou-se que os requisitos impostos pela organização possuem um caráter de alto nível, ficando a cargo do desenvolvedor da equipe a definição de muitas escolhas importantes de projeto, as quais impactam diretamente no desempenho do veículo na competição. Além disso, por vezes as regras são ambíguas, fato que ocasiona no aumento de tempo de desenvolvimento devido a intensa consulta a fóruns de dúvidas, relatos de outras equipes e mídias sociais.

A realização de cada módulo apresentado no trabalho passou por um extenso e criterioso processo de desenvolvimento. O design inicial fora realizado pelo autor, posteriormente consultando o professor orientador, o time de eletrônica da equipe, o diretor de projetos da área elétrica e algumas eventuais dúvidas com membros de equipes de outras universidades. Tal fato propiciou uma melhor transparência aos membros do projeto e uma forma de mitigação de eventuais erros apurada, já que os mesmos poderiam acarretar em consequências para a e-Power como um todo.

Devido as impossibilidades da atual situação, somente parte dos circuitos foi testada em bancada, o que impossibilitou a comparação com os valores teóricos e simulados. Sugere-se então, para trabalhos futuros, a finalização das simulações restantes, o teste dos demais circuitos utilizando placas de circuito impresso, elaboração de um chicote em bancada e instalação no carro, de forma a detectar eventuais problemas mecânicos com os cases, suportes e soldas mal realizadas. Também é interessante a elaboração de um manual de serviço da parte elétrica, a fim de achar e verificar falhas nos componentes, estados e tensões nos nós dos circuitos.

Vale também ressaltar que a diminuição do teor prático do trabalho ocasionou em uma maior exploração do firmware de uma das ECUs do veículo. Verificou-se que a IDE STM32 Cube é uma ferramenta relativamente fácil de trabalhar. O ambiente em

diagrama de blocos proporcionado para o desenvolvimento é intuitivo, eliminando parte das barreiras que a escrita de código pode impor a utilizadores leigos no assunto.

## 7 Apêndice A

```

/* Includes ----- */
include "main.h"

/* Private includes ----- */ /* USER CODE
BEGIN Includes */
include "stdbool.h"

include "Differential_Code.c"

#include "math.h"

/* USER CODE END Includes */

/* Private typedef ----- */ /* USER CODE
BEGIN PTD */

/* USER CODE END PTD */

/* Private define ----- */ /* USER CODE
BEGIN PD */ /* USER CODE END PD */

/* Private macro ----- */ /* USER CODE
BEGIN PM */

/* USER CODE END PM */

/* Private variables ----- */ ADC_HandleTypeDef
hadc1; ADC_HandleTypeDef hadc2;

FDCAN_HandleTypeDef hfdcan1;

IWDG_HandleTypeDef hiwdg;

TIM_HandleTypeDef htim1; TIM_HandleTypeDef htim2;

/* USER CODE BEGIN PV */ bool Ready_to_Drive_FLAG = false;

bool Implausibility_Flag = false;

bool Tractive_System_ON_FLAG = false; //flag para saber se o sistema de tracao
esta ativo

uint32_t Brake_value[10] = 0;

uint32_t APPS_value[10] = 0; //10 for the APPS1 sensor

uint32_t APPS2_value[10] = 0; //10 for the APPS1 sensor

uint32_t buffer[3];

```

```

uint8_t RxData[16];

uint8_t RxDataExpected[] = 0x10, 0x32, 0x54, 0x76, 0x98, 0x00, 0x11, 0x22, 0x33,
0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0x00;

uint8_t ErrorMessage[] = 0x00, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00;

uint8_t L = 1.7;

uint8_t D = 1.19;

float Rotations[2] = 0;

volatile float SWA_value_Average;

volatile float Brake_value_Average;

volatile float APPS1_value_Average;

volatile float APPS2_value_Average;

volatile float APPS_Result;

volatile float SWA_Result;

volatile uint32_t Right_Motor_Rotation;

volatile uint32_t Left_Motor_Rotation;

static FDCAN_RxHeaderTypeDef RxMessage;

static FDCAN_TxHeaderTypeDef TxMessage;

/* USER CODE END PV */

/* Private function prototypes _____*/ void System-
Clock_Config(void);

static void MX_GPIO_Init(void);

static void MX_FDCAN1_Init(void);

static void MX_ADC1_Init(void);

static void MX_TIM1_Init(void);

static void MX_TIM2_Init(void);

static void MX_ADC2_Init(void);

static void MX_IWDG_Init(void);

/* USER CODE BEGIN PFP */

void (*Pos_pointer)();

//prototypes:

```

```

void Init_State(); //State after the turn-on, waiting for the ready to drive start;
void Comm_State(); //State which the car is on the ready to drive mode;
//void Error_State(); //State which whenever error occurs.
void CAN_Filter_Config(void);
/* USER CODE END PFP */
/* Private user code _____ */ /* USER CODE
BEGIN 0 */
/* USER CODE END 0 */
/** * @brief The application entry point. * @retval int */ int main(void) /* USER
CODE BEGIN 1 */
/* USER CODE END 1 */
/* MCU Configuration _____ */
/* Reset of all peripherals, Initializes the Flash interface and the SysTick. */
HAL_Init();
/* USER CODE BEGIN Init */
HAL_FDCAN_Start(&hfdcan1);
/* USER CODE END Init */
/* Configure the system clock */
SystemClock_Config();
/* USER CODE BEGIN SysInit */
/* USER CODE END SysInit */
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_FDCAN1_Init();
MX_ADC1_Init();
MX_TIM1_Init();
MX_TIM2_Init();
MX_ADC2_Init();
MX_IWDG_Init();
/* USER CODE BEGIN 2 */

```

```

Pos_pointer = Init_State;
HAL_ADC_Start_IT(&hadc1);
HAL_ADC_Start_IT(&hadc2);
// HAL_FDCAN_Init(&hfdcan);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
  (*Pos_pointer)();
  HAL_Delay(20);
  HAL_IWDG_Refresh(
hiwdg); //Watchdog refresh
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */ /* USER CODE END 3 */

void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = 0;
  RCC_ClkInitTypeDef RCC_ClkInitStruct = 0;
  RCC_PeriphCLKInitTypeDef PeriphClkInit = 0;

  /** Configure the main internal regulator output voltage */
  HAL_PWREx_ControlVoltageScaling(PWR_REGULATOR
_VOLTAGE_SCALE1_BOOST);

  /** Initializes the CPU, AHB and APB busses clocks */
  RCC_OscInitStruct.OscillatorType =
RCC_OSCILLATORTYPE_HSI|RCC_OSCILLATORTYPE_LSI;
  RCC_OscInitStruct.HSISState = RCC_HSI_ON;
  RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
  RCC_OscInitStruct.LSISState = RCC_LSI_ON;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
  RCC_OscInitStruct.PLL.PLLM = RCC_PLLM_DIV4;
  RCC_OscInitStruct.PLL.PLLN = 85;

```

```

RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
RCC_OscInitStruct.PLL.PLLQ = RCC_PLLQ_DIV2;
RCC_OscInitStruct.PLL.PLLR = RCC_PLLR_DIV2;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK) Error_Handler();

/** Initializes the CPU, AHB and APB busses clocks */
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|
RCC_CLOCKTYPE_SYSCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV16;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV16;
if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_8) !=
HAL_OK)
    Error_Handler();

/** Initializes the peripherals clocks */
PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_ADC12|
RCC_PERIPHCLK_FDCAN;
PeriphClkInit.FdcanClockSelection = RCC_FDCANCLKSOURCE_PCLK1;
PeriphClkInit.Adc12ClockSelection = RCC_ADC12CLKSOURCE_SYSCLK;
if (HAL_RCCEX_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
    Error_Handler();

/** * @brief ADC1 Initialization Function
 * @param None
 * @retval None */ static void MX_ADC1_Init(void)
/* USER CODE BEGIN ADC1_Init 0 */
/* USER CODE END ADC1_Init 0 */
ADC_MultiModeTypeDef multimode = 0;
ADC_ChannelConfTypeDef sConfig = 0;

```

```

/* USER CODE BEGIN ADC1_Init 1 */
/* USER CODE END ADC1_Init 1 */

/** Common config */
hadc1.Instance = ADC1;

hadc1.Init.ClockPrescaler = ADC_CLOCK_ASYNC_DIV8;

hadc1.Init.Resolution = ADC_RESOLUTION_12B;

hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;

hadc1.Init.GainCompensation = 0;

hadc1.Init.ScanConvMode = ADC_SCAN_ENABLE;

hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;

hadc1.Init.LowPowerAutoWait = DISABLE;

hadc1.Init.ContinuousConvMode = ENABLE;

hadc1.Init.NbrOfConversion = 2;

hadc1.Init.DiscontinuousConvMode = DISABLE;

hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;

hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;

hadc1.Init.DMAContinuousRequests = DISABLE;

hadc1.Init.Overrun = ADC_OVR_DATA_OVERWRITTEN;

hadc1.Init.OversamplingMode = DISABLE;

if (HAL_ADC_Init(&hadc1) != HAL_OK)
    Error_Handler();

/** Configure the ADC multi-mode
 */
multimode.Mode = ADC_MODE_INDEPENDENT;

if (HAL_ADCEx_MultiModeConfigChannel(&hadc1, &multimode) != HAL_OK)
    Error_Handler(); /** Configure Regular Channel */

sConfig.Channel = ADC_CHANNEL_2;

sConfig.Rank = ADC_REGULAR_RANK_1;

sConfig.SamplingTime = ADC_SAMPLETIME_24CYCLES_5;

sConfig.SingleDiff = ADC_SINGLE_ENDED;

sConfig.OffsetNumber = ADC_OFFSET_NONE;

```



```

sConfig.Offset = 0;

if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
    Error_Handler();

/** Configure Regular Channel */
sConfig.Channel = ADC_CHANNEL_3;
sConfig.Rank = ADC_REGULAR_RANK_2;
if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
    Error_Handler();

/* USER CODE BEGIN ADC1_Init 2 */
/* USER CODE END ADC1_Init 2 */

/** * @brief ADC2 Initialization Function * @param None * @retval None */
static void MX_ADC2_Init(void)
{
    /* USER CODE BEGIN ADC2_Init 0 */
    /* USER CODE END ADC2_Init 0 */
    ADC_ChannelConfTypeDef sConfig = 0;

    /* USER CODE BEGIN ADC2_Init 1 */
    /* USER CODE END ADC2_Init 1 */

    /** Common config
    */
    hadc2.Instance = ADC2;

    hadc2.Init.ClockPrescaler = ADC_CLOCK_ASYNC_DIV8;

    hadc2.Init.Resolution = ADC_RESOLUTION_12B;

    hadc2.Init.DataAlign = ADC_DATAALIGN_RIGHT;

    hadc2.Init.GainCompensation = 0;

    hadc2.Init.ScanConvMode = ADC_SCAN_DISABLE;

    hadc2.Init.EOCSelection = ADC_EOC_SINGLE_CONV;

    hadc2.Init.LowPowerAutoWait = DISABLE;

    hadc2.Init.ContinuousConvMode = ENABLE;

    hadc2.Init.NbrOfConversion = 1;

    hadc2.Init.DiscontinuousConvMode = DISABLE;

    hadc2.Init.ExternalTrigConv = ADC_SOFTWARE_START;

```

```

hadc2.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
hadc2.Init.DMAContinuousRequests = DISABLE;
hadc2.Init.Overrun = ADC_OVR_DATA_OVERWRITTEN;
hadc2.Init.OversamplingMode = DISABLE;
if (HAL_ADC_Init(&hadc2) != HAL_OK)
    Error_Handler();

/** Configure Regular Channel */
sConfig.Channel = ADC_CHANNEL_1;
sConfig.Rank = ADC_REGULAR_RANK_1;
sConfig.SamplingTime = ADC_SAMPLETIME_2CYCLES_5;
sConfig.SingleDiff = ADC_SINGLE_ENDED;
sConfig.OffsetNumber = ADC_OFFSET_NONE;
sConfig.Offset = 0;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
    Error_Handler();

/* USER CODE BEGIN ADC2_Init 2 */
/* USER CODE END ADC2_Init 2 */

/** * @brief FDCAN1 Initialization Function
 * @param None * @retval None */ static void MX_FDCAN1_Init(void)
/* USER CODE BEGIN FDCAN1_Init 0 */
/* USER CODE END FDCAN1_Init 0 */
/* USER CODE BEGIN FDCAN1_Init 1 */
/* USER CODE END FDCAN1_Init 1 */
hfdcan1.Instance = FDCAN1;
hfdcan1.Init.ClockDivider = FDCAN_CLOCK_DIV1;
hfdcan1.Init.FrameFormat = FDCAN_FRAME_FD_NO_BRS;
hfdcan1.Init.Mode = FDCAN_MODE_NORMAL;
hfdcan1.Init.AutoRetransmission = DISABLE;
hfdcan1.Init.TransmitPause = DISABLE;
hfdcan1.Init.ProtocolException = DISABLE;

```

```

    hfdcan1.Init.NominalPrescaler = 1;
    hfdcan1.Init.NominalSyncJumpWidth = 1;
    hfdcan1.Init.NominalTimeSeg1 = 4;
    hfdcan1.Init.NominalTimeSeg2 = 2;
    hfdcan1.Init.DataPrescaler = 1;
    hfdcan1.Init.DataSyncJumpWidth = 1;
    hfdcan1.Init.DataTimeSeg1 = 1;
    hfdcan1.Init.DataTimeSeg2 = 1;
    hfdcan1.Init.StdFiltersNbr = 0;
    hfdcan1.Init.ExtFiltersNbr = 0;
    hfdcan1.Init.TxFifoQueueMode = FDCAN_TX_FIFO_OPERATION;
    if (HAL_FDCAN_Init(&hfdcan1) != HAL_OK) Error_Handler(); /* USER
CODE BEGIN FDCAN1_Init 2 */
    TxMessage.Identifier = 0x111;
    TxMessage.IdType = FDCAN_STANDARD_ID;
    TxMessage.TxFrameType = FDCAN_DATA_FRAME;
    TxMessage.DataLength = FDCAN_DLC_BYTES_8;
    TxMessage.ErrorStateIndicator = FDCAN_ESI_ACTIVE;
    TxMessage.BitRateSwitch = FDCAN_BRS_ON;
    TxMessage.FDFormat = FDCAN_FD_CAN;
    TxMessage.TxEventFifoControl = FDCAN_NO_TX_EVENTS;
    TxMessage.MessageMarker = 0;
    /* USER CODE END FDCAN1_Init 2 */
    /** * @brief IWDG Initialization Function * @param None * @retval None */
static void MX_IWDG_Init(void)
{
    /* USER CODE BEGIN IWDG_Init 0 */
    /* USER CODE END IWDG_Init 0 */
    /* USER CODE BEGIN IWDG_Init 1 */
    /* USER CODE END IWDG_Init 1 */
    hiwdg.Instance = IWDG;

```

```

hiwdg.Init.Prescaler = IWDG_PRESCALER_64;

hiwdg.Init.Window = 499;

hiwdg.Init.Reload = 499;

if (HAL_IWDG_Init(&hiwdg) != HAL_OK)

Error_Handler(); /* USER CODE BEGIN IWDG_Init 2 */

/* USER CODE END IWDG_Init 2 */

/** * @brief TIM1 Initialization Function * @param None * @retval None */ static
void MX_TIM1_Init(void)

/* USER CODE BEGIN TIM1_Init 0 */

/* USER CODE END TIM1_Init 0 */

TIM_ClockConfigTypeDef sClockSourceConfig = 0;

TIM_MasterConfigTypeDef sMasterConfig = 0;

TIM_BreakDeadTimeConfigTypeDef sBreakDeadTimeConfig = 0;

/* USER CODE BEGIN TIM1_Init 1 */

/* USER CODE END TIM1_Init 1 */

htim1.Instance = TIM1;

htim1.Init.Prescaler = 21250;

htim1.Init.CounterMode = TIM_COUNTERMODE_UP;

htim1.Init.Period = 50;

htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;

htim1.Init.RepetitionCounter = 0;

htim1.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;

if (HAL_TIM_Base_Init(&htim1) != HAL_OK)

Error_Handler();

sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;

if (HAL_TIM_ConfigClockSource(&htim1, &sClockSourceConfig) != HAL_OK)

Error_Handler();

sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;

sMasterConfig.MasterOutputTrigger2 = TIM_TRGO2_RESET;

sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;

```

```

    if (HAL_TIMEx_MasterConfigSynchronization(&htim1, &sMasterConfig) != HAL_OK)
        Error_Handler();

    sBreakDeadTimeConfig.BreakAFMode = TIM_BREAK_AFMODE_INPUT;
    sBreakDeadTimeConfig.Break2AFMode = TIM_BREAK_AFMODE_INPUT;
    if (HAL_TIMEx_ConfigBreakDeadTime(&htim1, &sBreakDeadTimeConfig) !=
        HAL_OK)
        Error_Handler();

    /* USER CODE BEGIN TIM1_Init 2 */

    /* USER CODE END TIM1_Init 2 */

    /** * @brief TIM2 Initialization Function * @param None * @retval None */ static
void MX_TIM2_Init(void)
{
    /* USER CODE BEGIN TIM2_Init 0 */

    /* USER CODE END TIM2_Init 0 */

    TIM_ClockConfigTypeDef sClockSourceConfig = 0;
    TIM_MasterConfigTypeDef sMasterConfig = 0;

    /* USER CODE BEGIN TIM2_Init 1 */

    /* USER CODE END TIM2_Init 1 */

    htim2.Instance = TIM2;
    htim2.Init.Prescaler = 21250;
    htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim2.Init.Period = 100;
    htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
        Error_Handler();

    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig) != HAL_OK)
        Error_Handler();

    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;

```

```

    if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) != HAL_OK)
Error_Handler();

/* USER CODE BEGIN TIM2_Init 2 */

/* USER CODE END TIM2_Init 2 */

/** * @brief GPIO Initialization Function * @param None * @retval None */ static
void MX_GPIO_Init(void)

{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOA, BrakeRangesOK_Pin
|Buzzer_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOB, LED_2_Pin
|LED_1_Pin|LD2_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin : Brake_ON_OFF_Pin */
    GPIO_InitStruct.Pin = Brake_ON_OFF_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(Brake_ON_OFF_GPIO_Port, &GPIO_InitStruct);

    /*Configure GPIO pin : Ready_to_Drive_Button_ON_OFF_Pin */
    GPIO_InitStruct.Pin = Ready_to_Drive_Button_ON_OFF_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(Ready_to_Drive_Button_ON_OFF_GPIO_Port,
&GPIO_InitStruct);

    /*Configure GPIO pins : BrakeRangesOK_Pin Buzzer_Pin */
    GPIO_InitStruct.Pin = BrakeRangesOK_Pin|Buzzer_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;

```

```

GPIO_InitStruct.Pull = GPIO_NOPULL;

GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;

HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/*Configure GPIO pin : LED_3_Pin */
GPIO_InitStruct.Pin = LED_3_Pin;

GPIO_InitStruct.Mode = GPIO_MODE_ANALOG;

GPIO_InitStruct.Pull = GPIO_NOPULL;

HAL_GPIO_Init(LED_3_GPIO_Port, &GPIO_InitStruct);

/*Configure GPIO pins : LED_2_Pin LED_1_Pin LD2_Pin */
GPIO_InitStruct.Pin = LED_2_Pin|LED_1_Pin|LD2_Pin;

GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;

GPIO_InitStruct.Pull = GPIO_NOPULL;

GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;

HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/* EXTI interrupt init*/

HAL_NVIC_SetPriority(EXTI9_5_IRQn, 3, 0);

HAL_NVIC_EnableIRQ(EXTI9_5_IRQn);

/* USER CODE BEGIN 4 */

//Function to calculate the rotation based on the APPS pressure and the SWA
angle void Ackermann_Equation(float APPS_Result, float SWA_Result, float *Rotations)
uint32_t Actual_Rot = APPS_Result*300;

Rotations[0] = Actual_Rot*(1 + ((D/2)*tan(SWA_Result))/L);
Rotations[1] = Actual_Rot*(1 - ((D/2)*tan(SWA_Result))/L);

//Detects Ready to drive button interrupt (GPIO_PIN_7)
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)

if((GPIO_Pin == GPIO_PIN_7) && (Tractive_System_ON_FLAG == true)
&&

(HAL_GPIO_ReadPin(Brake_ON_OFF_GPIO_Port,Brake_ON_OFF_Pin) ==
GPIO_PIN_SET) && (Ready_to_Drive_FLAG == false))

HAL_TIM_Base_Start_IT(&htim1);

else __NOP();

```

```

void Debouncing_Check() //If Ready to Drive button and brake remain pres-
sed..Set the Ready to Drive Mode flag to True.

if((HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_7) == GPIO_PIN_SET) &&
   (HAL_GPIO_ReadPin(Brake_ON_OFF_GPIO_Port,Brake_ON_OFF_Pin) ==
GPIO_PIN_SET)) HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_RESET);
Ready_to_Drive_FLAG = true; else (*Pos_pointer)();

static uint32_t BufferCmp8b(uint8_t* pBuffer1, uint8_t* pBuffer2, uint16_t
BufferLength)

while(BufferLength-)

if(*pBuffer1 != *pBuffer2)

return 1;

pBuffer1++;

pBuffer2++;

return 0;

void Init_State(void) //Blinks the init state indicator LED

HAL_GPIO_TogglePin(LED_1_GPIO_Port, LED_1_Pin);

while (HAL_FDCAN_GetRxFifoFillLevel(&hfdcan1, FDCAN_RX_FIFO0) < 2)

/* Receive Rx messages from RX FIFO0 */ HAL_FDCAN_GetRxMessage(&hfdcan1,
FDCAN_RX_FIFO0, &RxMessage, RxData);

HAL_FDCAN_GetRxMessage(&hfdcan1, FDCAN_RX_FIFO0, &RxMessage,
&RxData[8]);

//Compare the received data (CAN) with the reference

if ((BufferCmp8b(RxDataExpected, RxData, 16) != 0) || (Ready_to_Drive_FLAG
== false))

Pos_pointer = Init_State;

else if (Ready_to_Drive_FLAG == true)

Pos_pointer = Comm_State;

//Turn on buzzer for 2 seconds

HAL_GPIO_WritePin(Buzzer_GPIO_Port, Buzzer_Pin, GPIO_PIN_SET);

HAL_Delay(2000);

HAL_GPIO_WritePin(Buzzer_GPIO_Port, Buzzer_Pin, GPIO_PIN_RESET);

//Check if the Ready to Drive Mode Flag is true.

```



```

void Comm_State(void)

//Blinks the comm state indicator LED
HAL_GPIO_TogglePin(LED_2_GPIO_Port, LED_2_Pin);

//Start the ADC's for reading the APPS's and SWA values
for( int i=0; i <10; i++)

HAL_ADC_Start(&hadc1);
HAL_ADC_Start(&hadc2);

HAL_ADC_PollForConversion(&hadc2, 100);

Brake_value_Average += 0.1*HAL_ADC_GetValue(&hadc2);

HAL_ADC_PollForConversion(&hadc1, 100);

APPS1_value_Average += 0.1*HAL_ADC_GetValue(&hadc1);

HAL_ADC_PollForConversion(&hadc1, 100);

APPS2_value_Average += 0.1*HAL_ADC_GetValue(&hadc1);

HAL_ADC_Stop (&hadc1);
HAL_ADC_Stop(&hadc2);

//Transform to voltage Brake_value_Average = (Brake_value_Average*3.3)/4095;
APPS1_value_Average = (APPS1_value_Average*3.3)/4095;
APPS2_value_Average = (APPS2_value_Average*3.3)/4095;

APPS2_value_Average = APPS2_value_Average/3; //Value used to remove
APPS2 gain

if(((APPS1_value_Average) == 0.0)
||((APPS2_value_Average) == 0.0)
||((APPS2_value_Average) > 3.29)||((APPS1_value_Average) > 3.29))

//If a short circuit or open circuit occurs... a flag is set
HAL_TIM_Base_Start_IT(&htim2);

if(((APPS1_value_Average - APPS2_value_Average) > 0.1)
||((APPS2_value_Average - APPS1_value_Average) > 0.1))

//If a deviation more than 10% occurs between the sensors... a flag is set
HAL_TIM_Base_Start_IT(&htim2);

APPS_Result = APPS1_value_Average + APPS2_value_Average;

```

```

    if(((Brake_value_Average) == 0.0)||((Brake_value_Average) > 3.29)) //If a
short circuit or open circuit occurs... a flag is set

    //GPIO to send the Brake On/Off signal to BSPD
    HAL_GPIO_WritePin(BrakeRangesOK
    _GPIO_Port, BrakeRangesOK_Pin, GPIO_PIN_RESET);
    HAL_TIM_Base_Start_IT(&htim2);
    else if(Brake_value_Average > 3)
    HAL_GPIO_WritePin(BrakeRangesOK
    _GPIO_Port, BrakeRangesOK_Pin, GPIO_PIN_SET);
    else
    HAL_GPIO_WritePin(BrakeRangesOK
    _GPIO_Port, BrakeRangesOK_Pin, GPIO_PIN_RESET);
    if(((Brake_value_Average) > 0.5)
    &&((APPS_Result) > 0.825))
    //If brake is pressed and apps > 25
    HAL_TIM_Base_Start_IT(&htim2);
    Ackermann_Equation(APPS_Result, SWA_Result, Rotations);
    HAL_FDCAN_Start(&hfdcan1);
    HAL_FDCAN_AddMessageToTxFifoQ(&hfdcan1, &TxMessage, &Rotations[0]);
    HAL_FDCAN_AddMessageToTxFifoQ(&hfdcan1, &TxMessage, &Rotations[1]);
    //HAL_FDCAN_Stop(&hfdcan1);
    void Implausibility_Check()

    if(((APPS1_value_Average - APPS2_value_Average) > 0.1)||((APPS2_value_Average
- APPS1_value_Average) > 0.1))
    //If a deviation more than 10% occurs between the sensors... a flag is set
    Pos_pointer = Error_Handler;
    if(((APPS1_value_Average) == 0.0)
    ||((APPS2_value_Average) == 0.0)
    ||((APPS2_value_Average) > 3.29)||((APPS1_value_Average) > 3.29))
    //If a short circuit or open circuit occurs... a flag is set

```

```

Pos_pointer = Error_Handler;

if(((Brake_value_Average) == 0.0)||((Brake_value_Average) > 3.29))

//If a short circuit or open circuit occurs... a flag is set

Pos_pointer = Error_Handler;

HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)

if(htim->Instance==TIM1)

Debouncing_Check();

if(htim->Instance==TIM2)

Implausibility_Check();

/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 *
 * @retval None
 */
void Error_Handler(void)
/* USER CODE BEGIN Error_Handler_Debug */
/* User can add his own implementation to report the HAL error return state */
//Blinks the error state indicator LED
HAL_GPIO_TogglePin(LED_3_GPIO_Port, LED_3_Pin);
HAL_FDCAN_AddMessageToTxFifoQ(&hfdcan1, &TxMessage, &ErrorMessage);
/* USER CODE END Error_Handler_Debug */
#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 *
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 *
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)

```

```
/* USER CODE BEGIN 6 */  
/* User can add his own implementation to report the file name and line number,  
tex: printf("Wrong parameters value: file  
/* USER CODE END 6 */  
endif /* USE_FULL_ASSERT */
```

## 8 Referências Bibliográficas

- [1] STEVE CORRIGAN, TEXAS INSTRUMENT, *Introduction to the Controller Area Network (CAN), Application Report*. SLOA101B, 2017.
- [2] J. A. COOK, J. S. FREUDENBERG, *Controller Area Network (CAN)*, EECS 461, 2008.
- [3] Robert Bosch GmbH, “*CAN Specification 2.0*,” 1991.
- [4] BARHAM, Matt. *Design and development of the Electrical Systems in an Electric Formula SAE race car*. New Zealand, 2017. 245 p.
- [5] Thomas Fuehrer, Robert Hugel, Florian Hartwich and Harald Weiler, “*FlexRay - The Communication System for Future Control Systems in Vehicles*,” SAE Paper 2003-01-0110, 2003.
- [6] *Compact Circular Connectors*. Souriau, 2012. Disponível em <<http://www.souriau.co.jp/material/8sta.pdf>>. Acesso em: 30 abril 2019.
- [7] *Steering angle sensor diagnostics. Know your parts*. Disponível em <<https://www.knowyourparts.com/technical-resources/suspension/steering-angle-sensor-diagnostics/>> . Acesso em: 24 Abri 2019.
- [8] *Insulation Monitoring*. Bender. Disponível em <[https://www.bender.de/en/products/insulation-monitoring/isometer\\_ir155-3203ir155-3204](https://www.bender.de/en/products/insulation-monitoring/isometer_ir155-3203ir155-3204)>. Acesso em: 3 Mar 2019.
- [9] *Insulation monitoring devices for fault detection and location*. Schneider Eletric. Disponível em <<https://www.schneider-electric.com/en/product-range/817-vigilohm/>>. Acesso em: 4 Mai 2019.
- [10] *Electrical System Form FSAE-E2018*. University of Wisconsin-Madison. Disponível em <[https://static1.squarespace.com/static/57e8888fc534a547699d733d/t/5bacf752652dea1a20d5517c/1538062172553/University\\_of\\_Wisconsin\\_Madison\\_Electric\\_2018\\_ESF\\_Submission\\_2.pdf](https://static1.squarespace.com/static/57e8888fc534a547699d733d/t/5bacf752652dea1a20d5517c/1538062172553/University_of_Wisconsin_Madison_Electric_2018_ESF_Submission_2.pdf)>. Acesso em: 1 Mar 2019.
- [11] BOHM, Augusto. *Electronic Differential for Electric Vehicles*. Porto Alegre, 2018.108 p.
- [12] *Fundamentals of Noise and Soud*. FAA. Disponível em <[https://www.faa.gov/regulationspolicies/policy\\_guidance/noise/basics/](https://www.faa.gov/regulationspolicies/policy_guidance/noise/basics/)>. Acesso em: 11 Mai 2019.
- [13] *Lightweight, compact and high-efficiency powertrain for electric vehicles EV*. Disponível em <[https://www.nissan-global.com/EN/TECHNOLOGY/OVERVIEW/e\\_powertrain.html](https://www.nissan-global.com/EN/TECHNOLOGY/OVERVIEW/e_powertrain.html)>. Acesso em 11 Mai 2019.

- [14] *Batteries for Electric Cars: Challenges, Opportunities and the Outlook for 2020*. BCG. Disponível em <<https://www.bcg.com/documents/file36615.pdf>>. Acesso em 11 Mai 2019.
- [15] HART, Daniel W. *Eletrônica de Potência: Análise e Projetos de Circuitos*. 1 ed. Porto Alegre: AMGH, 2011.
- [16] DEJUN YIN, DANFENG SHAN AND JIA-SHENG HU, *A Study on the Control Performance of Electronic Differential System for Four-Wheel Drive Electric Vehicles*, 3, 2017.
- [17] Novas tecnologias: Carros atuais possuem até 100 sensores a bordo. Disponível em <<https://quatorrodas.abril.com.br/noticias/novas-tecnologias-carros-atuais-tem-ate-100-sensores-a-bordo/>>. Acesso em 3 Set 2019.
- [18] Módulo CAN MCP2515 TJA1050. Disponível em <<https://www.eletrogate.com/modulo-can-bus-mcp2515-tja1050>>. Acesso em 4 Set 2019.
- [19] *ACS718: High Isolation Linear Current Sensor IC with 850 Current Conductor*. Disponível em <<https://datasheetspdf.com/pdf/1410949/Allegro/ACS718/1>>. Acesso em 7 Set 2019.
- [20] *Arduino Mega Overview*. Disponível em <<https://store.arduino.cc/usa/mega-2560-r3>>. Acesso em 6 Out 2019.
- [21] Chave fim de curso com rolete. Disponível em <<https://www.eletrogate.com/chave-fim-de-curso-com-rolete>>. Acesso em 6 Out 2019.
- [22] *Electric AMZ Racing*. Disponível em <<http://electric.amzracing.ch/>>. Acesso em 8 Out 2019.
- [23] SENSOR DE POSIÇÃO DA BORBOLETA (TPS). Disponível em <<https://www.ds.ind.br/pt/produtos/sensor-de-posicao-da-borboleta-tps/1901>>. Acesso em 23 Ago 2020.
- [24] SENSOR DE PRESSÃO PS1500PSI FUELTECH (0-102BAR). Disponível em <<https://fueltech.com.br/products/sensor-de-pressao-ps1500psi-fueltech>>. Acesso em 2 Set 2020.
- [25] *CAN FD Transceiver with Silent Mode*. Disponível em <<https://ww1.microchip.com/downloads/en/DeviceDoc/20005533A.pdf>>. Acesso em 7 Set 2020.
- [26] Motor elétrico automotivo: entenda as diferenças e vantagens. Disponível em <<https://chiptronic.com.br/blog/motor-eletrico-automotivo-entenda-as-diferencas-e-vantagens>>:text=Alta%20efici%C3%Aancia%20energ%C3%A9tica,realidade%20causa%20a%20inda%20mais%20desapontamento.> Acesso em 25 Nov 2020.